

Dual Satellite Coverage using Particle Swarm Optimization

Juan A. Ojeda Romero

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in
partial fulfillment of the requirements for the degree of

Master of Science
In
Aerospace Engineering

Michael K. Philen, Chair
Gregory D. Earle
Cornel Sultan

October 3, 2014
Blacksburg, Virginia

Keywords: Particle Swarm Optimization, space weather, aurora borealis, orbital mechanics,
satellite coverage

Copyright© 2014, Juan A. Ojeda Romero

Dual Satellite Coverage using Particle Swarm Optimization

Juan A. Ojeda Romero

ABSTRACT

A dual satellite system in a Low Earth Orbit, LEO, would be beneficial to study the electromagnetic occurrences in the magnetosphere and their contributions to the development of the aurora events in the Earth's lower atmosphere. An orbit configuration is sought that would increase the total time that both satellites are inside the auroral oval. Some additional objectives include minimizing the total fuel cost and the average angle between the satellites' radius vectors. This orbit configuration is developed using a series of instantaneous burns applied at each satellite's perigee. An analysis of the optimal solutions generated by a Particle Swarm Optimization method is completed using a cost function with different weights for the time, fuel, and angle terms. Three different scenarios are presented: a single burn case, a double burn case, and a four burn case. The results are calculated using two different orbital mechanics models: an unperturbed two-body simulation and a two-body simulation with added Earth's equatorial bulge effects. It is shown that the added perturbation reduces the total event time in the optimal solutions generated. Specific weights for the cost function are recommended for further studies.

Acknowledgments

First, I would like to thank all the friends I have made during my time as an undergraduate and graduate student at Virginia Tech. They gave me the support I needed while I was adjusting to life away from home. I also would like to express my gratitude to Dr. Gregory Earle and Dr. Cornel Sultan for participating as my committee members and for all their support throughout my graduate studies. I wish to express my most sincere gratitude to Dr. Michael Philen for all his mentorship and support this past year. This research was only possible with his guidance and patience.

Finally, I would like to express my gratitude and love to my parents, Juan and Maria. Thank you for giving me the courage and perseverance to follow my dreams.

Table of Contents

1. INTRODUCTION.....	1
1.1 SPACE SCIENCE.....	1
1.1.1 Solar Wind and Solar Events	2
1.1.2 Geomagnetic Field and Magnetosphere.....	3
1.1.3 Aurora formation	4
1.2 SPACE WEATHER SATELLITES	5
1.3 MOTIVATION.....	8
1.4 RESEARCH APPROACH	8
1.5 THESIS OUTLINE	9
2. ORBITAL DYNAMICS	10
2.1 FRAME OF REFERENCE	10
2.2 KEPLERIAN TWO BODY PROBLEM	11
2.3 J_2 EFFECT	15
2.4 NUMERICAL INTEGRATION.....	18
2.5 SUMMARY	19
3. PARTICLE SWARM OPTIMIZATION	20
3.1 BRIEF INTRODUCTION	20
3.2 OPTIMIZATION PROBLEM	24
3.2.1 Initial conditions	25
3.2.2 Cost Function	27
3.2.3 Theoretical Event Time and Angle	35
3.3 PSO ALGORITHM	41
4. RESULTS AND DISCUSSION	46
4.1 UNPERTURBED TWO-BODY MOTION	46
4.1.1 Single Burn Optimization	46
4.1.2 Double Burn Optimization.....	56
4.1.3 Four Burn Optimization.....	61
4.1.4 Summary	63
4.2 TWO-BODY MOTION WITH J_2 EFFECT	64
4.2.1 Single Burn Optimization	64
4.2.2 Double Burn Optimization.....	68
4.2.3 Four Burn optimization.....	71
4.2.4 Summary	73
5. CONCLUSIONS.....	75

5.1 FUTURE WORK	76
BIBLIOGRAPHY	78
APPENDICES	80
APPENDIX A. TABLES OF OPTIMAL SOLUTIONS	80
APPENDIX B. PSO MATLAB CODE	86

List of Figures

Figure 1.1: Magnetosphere generated by the interaction between the Earth and the solar wind. Earth's Magnetosphere. 2014; Available from: http://www-ssc.igpp.ucla.edu/personnel/russell/papers/magsphere/msphere06.gif . Used under fair use, 2014	3
Figure 1.2: Schematic of THEMIS mission satellite layout, <i>THEMIS satellites</i> . 2014; Available from: http://www.nasa.gov/images/content/164101main_araural_disruption_med.jpg . Used under fair use, 2014	7
Figure 2.1: Earth Centered Inertial, ECI, reference frame. The latitude and longitude are defined in this figure, where the motion of a satellite is given in red.	11
Figure 2.2: Classical orbital elements, Vallado, D.A. and W.D. McClain, <i>Fundamentals of Astrodynamics and Applications</i> . 2007: Springer. Used under fair use, 2014	15
Figure 2.3: J_2 effect on the regression of the Right Ascension of the Ascending Node. Gurfil, P., <i>Modern Astrodynamics</i> . 2006: Elsevier Science. Used under fair use, 2014	18
Figure 2.4: J_2 effect on the regression of the line of apsides. Gurfil, P., <i>Modern Astrodynamics</i> . 2006: Elsevier Science. Used under fair use, 2014	18
Figure 3.1: Initial orbit of satellites. Red: satellite 1. Blue: satellite 2.	25
Figure 3.2: Effect of instantaneous burns.	28
Figure 3.3: Scenario 2 – calculation of t_a is made with radius vectors from satellite 1....	32
Figure 3.4: Scenario 3 - calculation of t_a is made with Eq. (3.23-3.26).....	32
Figure 3.5: Calculate event times and average angle.....	34
Figure 3.6: PQW Reference Frame.....	35
Figure 3.7: Eccentric Anomaly.	37
Figure 3.8: Cost terms for case of $\alpha=0.8$ and $\beta=0.5$	45
Figure 4.1: Satellite 1 single burn cost.....	47
Figure 4.2: Satellite 2 single burn cost.....	48
Figure 4.3: Cost function terms for satellite 1 single burn at $\alpha=0.5$	49
Figure 4.4: Cost function terms for satellite 2 single burn at $\alpha=0.5$	49
Figure 4.5: Single burn optimization time. Left: satellite 1 burn only. Right: satellite 2 burn only.	50
Figure 4.6: Events and Time. Left: satellite 1 burn only. Right: satellite 2 burn only.	51
Figure 4.7: Single burn optimization with no average angle ($\beta=0$). The optimal solutions are labeled using markers and α ranges only apply for numerical solutions. Left: satellite 1. Right: satellite 2.	52
Figure 4.8: Satellite 1 single burn cost with $\beta=0.5$	53
Figure 4.9: Satellite 2 single burn cost with $\beta=0.5$	53

Figure 4.10: Cost function terms for $\alpha=0.5$ and $\beta=0.5$. Left: satellite 1 single burn. Right: satellite 2 single burn.	54
Figure 4.11: Average angle calculation for single burn case with $\alpha=0.5$ and $\beta=0.5$. Left: satellite 1 single burn. Right: satellite 2 single burn.	55
Figure 4.12: Single burn optimization with average angle included ($\beta=0.5$). The optimal solutions are labeled using markers and α ranges only apply for numerical solutions. Left: satellite 1. Right: satellite 2.	56
Figure 4.13: Average angle comparison of optimal solutions for single burn optimization with unperturbed motion.....	56
Figure 4.14: Surface of cost function with no average angle ($\beta=0$).	57
Figure 4.15: Left: velocity term. Right: total event time term.	57
Figure 4.16: Pareto front for double burn optimization with no average angle ($\beta=0$). Optimal solutions are labeled in red with their appropriate α ranges.	58
Figure 4.17: Cost value for double burn optimization with average angle ($\beta=0.5$).	59
Figure 4.18: Function terms for $\alpha=0.6$. Left: Dimensionless average angle term. Right: Dimensionless cost term.	59
Figure 4.19: Pareto front of double burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled in red with corresponding α ranges.	60
Figure 4.20: Average angle of optimal solutions with ($\beta=0.5$) and without ($\beta=0$) its inclusion in the double burn optimization.	61
Figure 4.21: Pareto front of four burn optimization case with no average angle included ($\beta=0$). Optimal solutions are labeled in red with corresponding α ranges.	62
Figure 4.22: Pareto front of four burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled in red with corresponding α ranges.	63
Figure 4.23: Average angle of optimal solutions with ($\beta=0.5$) and without ($\beta=0$) its inclusion in the four burn optimization.....	63
Figure 4.24: Cost of J_2 effect single burn optimization. Left: satellite 1 burn only. Right: satellite 2 burn only.....	65
Figure 4.25: Cost terms single burn optimization with $\alpha=0.5$. Left: satellite 1 burn only. Right: satellite 2 burn only.....	65
Figure 4.26: J_2 effect single burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply to J_2 effect solutions. Left: satellite 1 burn only. Right: satellite 2 burn only	66
Figure 4.27: J_2 effect single burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply to J_2 effect solutions. Left: satellite 1. Right: satellite 2.	67
Figure 4.28: Average angle of optimal solutions comparison of single burn optimization with J_2 effect. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.....	68
Figure 4.29: Cost surface with variable alpha for $\beta=0$	69
Figure 4.30: Cost surface with variable alpha for $\beta=0.5$	69
Figure 4.31: Pareto front of J_2 effect double burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply for J_2 effect optimization.	70

Figure 4.32: Pareto front of J_2 effect double burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply for J_2 effect optimization.....	71
Figure 4.33: Angle comparison of optimal solutions of double burn optimization with J_2 effect. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.	71
Figure 4.34: Pareto front of J_2 effect four burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply for J_2 optimization.....	72
Figure 4.35: Pareto front of J_2 effect four burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply for J_2 optimization.....	73
Figure 4.36: Angle comparison of optimal solutions of J_2 effect four burn optimization. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.	73
Figure 4.37: J_2 effect on satellite 1 under initial conditions.	74
Figure 4.38: J_2 effect on satellite 2 under initial conditions.	74

List of Tables

Table 2.1: Keplerian orbital elements.....	14
Table 3.1: Initial orbital elements of satellites.....	25
Table 3.2: Particle Swarm Optimization constants.....	44
Table A.1: Single burn optimization for satellite 1 with $\beta=0$	80
Table A.2: Single burn optimization for satellite 1 with $\beta=0.5$	80
Table A.3: Single burn optimization for satellite 2 with $\beta=0$	81
Table A.4: Single burn optimization for satellite 2 with $\beta=0.5$	81
Table A.5: Double burn optimization with $\beta=0$	81
Table A.6: Double burn optimization with $\beta=0.5$	82
Table A.7: Four burn optimization with $\beta=0$	82
Table A.8: Four burn optimization with $\beta=0.5$	83
Table A.9: J_2 effect single burn optimization satellite 1 with $\beta=0$	83
Table A.10: J_2 effect single burn optimization satellite 1 with $\beta=0.5$	83
Table A.11: J_2 effect single burn optimization satellite 2 with $\beta=0$	84
Table A.12: J_2 effect single burn optimization satellite 2 with $\beta=0.5$	84
Table A.13: J_2 effect double burn optimization with $\beta=0$	84
Table A.14: J_2 effect double burn optimization with $\beta=0.5$	85
Table A.15: J_2 effect Four burn optimization with $\beta=0$	85
Table A.16: J_2 effect Four burn optimization with $\beta=0.5$	85

Chapter 1

Introduction

Space weather is an emerging field of study due to its impact on satellite systems in orbit and electrodynamic systems on Earth. Most of the adverse effects on satellites occur from the impact of charged particles on the satellite's surface in Low Earth Orbit, LEO. The concentration and composition of these charged particles is dependent on processes from the Sun's surface. The interaction between the material expelled from the result of these processes and particles in the ionosphere create the aurora displays. An increasing topic of interest in the area of space weather is the study of these specific processes that cause the formation of the aurora, more commonly known as the bright lights seen at high latitudes. In this section, a brief description on some selected topics in the field of space weather is given, including the aurora. This is followed by a description of some of the satellite missions that have studied the charged particle interactions inside the Earth's ionosphere and magnetosphere.

1.1 Space Science

Space weather is the study of processes caused by highly energized particles emitted from the Sun's solar wind. These processes occur near the outer edge of the atmosphere and affect systems both on the Earth and in orbit and are normally caused by the interaction between the

Earth and Sun, aside from gravitational attraction. The following is a brief description into the space weather science around the Earth and some motives for research into the auroral regions.

1.1.1 Solar Wind and Solar Events

The interaction between the Earth and the Sun is not limited to gravitational attraction, but also includes the exchange between highly charged plasma. The fusion process that occurs in the Sun creates highly energized ionized particles that are expelled from the surface of the Sun at high speeds. These fully ionized particles travel at supersonic speeds (≈ 400 km/s) caused by a combination of heating, compression, and expansion due to the gravitational field around the Sun and the subsequent fusion reaction [1]. This material of ionized particles, or plasma, is also called the solar wind. The Sun also has a very complex magnetic field with certain areas of surprisingly high magnetic field magnitude called sunspots. As the Sun expels the plasma material that makes up the solar wind, some of the magnetic field lines along the surface are trapped in these plasma discharges and travel outward as a part of the solar wind. Kelley [1] also notes that the plasma that travels as part of the solar wind carries both kinetic and electrical energy. In turn this energy eventually finds its way towards the lower atmosphere, where it has been known to “power” the aurora by exciting particles in the lower atmosphere. The release of the solar wind from the Sun is a continuous process that is caused by the Sun’s fusion reaction and intense surface heat, however the energy of the ionized plasma that arrives at Earth may vary. Occasionally, there may periods where certain areas of the Sun’s surface may expel a large amount of charged particles, these are called solar flares [2]. These are highly energized plasma events that have a variety of impacts on satellite performance and may adversely affect electronic and communication systems on Earth.

1.1.2 Geomagnetic Field and Magnetosphere

The Earth's magnetic field may be simply described as a simple dipole tilted about 11° from the geographic North Pole. In this model, the magnetic field points away from the Earth's surface in the northern hemisphere and towards the surface in the southern hemisphere. In reality, the magnitude and shape of the Earth's magnetic field is more complex and the geomagnetic poles have been known to change position over time. However, near the Earth, the dipole effect is more prevalent and the rate of change of the geomagnetic pole position will not be used in this thesis. Due to the interaction between the Earth and the Sun, the magnetic field lines around the Earth are not as simple as a magnetic dipole [1]. The solar wind causes some of the magnetic field lines to stretch or compress depending on the position of the Sun relative to the Earth, see Figure 1.1.

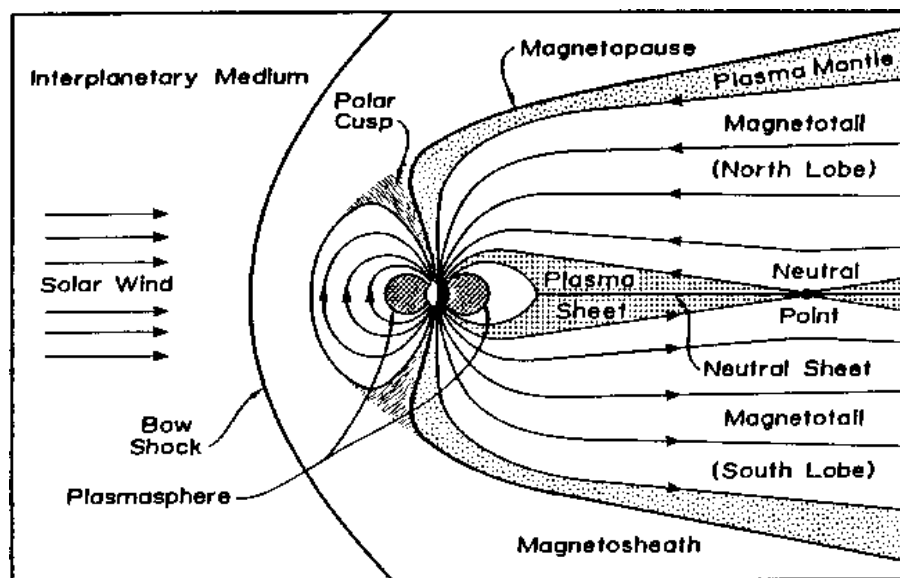


Figure 1.1: Magnetosphere generated by the interaction between the Earth and the solar wind, Earth's Magnetosphere. 2014; Available from: <http://www-ssc.igpp.ucla.edu/personnel/russell/papers/magsphere/msphere06.gif>. Used under fair use, 2014

The magnetosphere is a region that surrounds the Earth which is created by the incoming solar wind and it is dominated by the Earth's magnetic field. Note that the solar wind is moving at supersonic speeds in the interplanetary medium, therefore a bow shock is formed in between the

Earth and solar wind. The region between this bow shock and the Earth is known as the magnetosheath, a region of slowly moving particles that are directly responsible for the shape of the magnetosphere. In the sunward side, there is a boundary that terminates the magnetosphere, this region is the magnetopause. Between the magnetopause and the Earth, the magnetic field lines are either compressed close to the Earth or stretched towards the anti-sunward side. The anti-sunward region of the magnetosphere is characterized by long stretched magnetic field lines that extend toward the magnetotail, the far end of the magnetosphere [1].

1.1.3 Aurora formation

The light-emitting, or discrete, aurora is the photon emission of neutral gas particles as they are ionized by highly-energized plasma from the Earth's ionosphere and the solar wind [1]. The auroral events caused by energized electrons entering the Earth's ionosphere can be classified into discrete and diffuse auroras [4]. The discrete aurora is associated with the Aurora Borealis in the northern hemisphere and the Aurora Australis in the southern hemisphere. The region of the atmosphere that is associated with the appearance of the aurora is known as the ionosphere, which is characterized by high concentrations of ions at an altitude range of 90-1000 km. There are two possible scenarios in which energized plasma from the solar wind may cause the aurora display. Plasma from the solar wind may be trapped inside one of the cusps near the poles caused by the solar wind, see Figure 1.1. The plasma that enters these cusps consists of electrons and ions that penetrate the ionosphere and precipitate into the lower atmosphere. The collision between these particles and the neutral gases in the lower atmosphere cause the auroral lights. The alternative method by which the aurora may be created is through magnetic reconnection that occurs in the anti-sunward region of the magnetosphere. Magnetic field lines from the sunward direction are stretched toward the tail of the magnetosphere, these field lines may reconnect and

drive energized electrons and ions back toward the Earth. The point where reconnection occurs can be seen at the neutral point in Figure 1.1. This event is responsible for part of the ionization in the ionosphere and the diffuse aurora, which are widespread emissions seen over the lower atmosphere not limited to the region over the poles [1]. The generation of the aurora is directly formed by the acceleration of the electrons in either of these two methods. More notably, the study of particles trapped in the polar cusps has been the objective of many satellite missions. Colpitts *et al.* has listed 3 regions of auroral electron acceleration in the polar regions: an upward region with downgoing "Inverted-V" electrons, a downward region with upgoing electrons, and the polar cap acceleration region (associated with alfvénic acceleration) [5].

1.2 Space Weather Satellites

The electromagnetic phenomena that occur in the magnetosphere and the lower atmosphere have been studied with the use of ground stations on Earth, primarily near the poles, and both satellites and sounding rockets. In particular, the use of sounding rockets and satellites allows in-situ data to be collected at a higher resolution and spatial rate. To study the acceleration of the auroral electrons in the upper atmosphere, the use of a satellite is preferred to that of a sounding rocket. While a sounding rocket has the capability for a higher spatial rate than satellites, they are not able to reach the altitudes in which most of the electron acceleration occurs. The area of interest in this thesis is an altitude region higher than 100 km and focuses on the geomagnetic latitude region over the auroral ovals. Data taken in this region have come from a number of satellite missions tasked with investigating the dynamic magnetic field, electric field, and ion concentration. This section covers a few of these mission and their scientific conclusions.

The Fast Auroral Snapshot (FAST) Mission was part of NASA's Small Explorer (SMEX) Satellite Initiative. The main focus of this program was to perform "Rapid, low cost, and highly

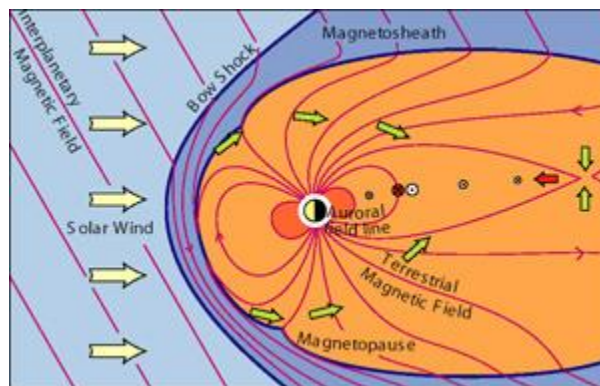
focused scientific investigations” with data measurement collected at a “high spatial and temporal resolution” [6] over the auroral ovals at altitudes of 4100 km over the geographic poles. Lee [4] suggests that the energized electrons found in the auroral oval come directly from the equatorial plasma sheet, see Figure 1.1. This conclusion was taken from data found in three conjugate events from the THEMIS and FAST satellites. These events occurred when each satellite was along the same magnetic field line. In addition, Colpitts *et al* [5] were able to observe three distinct types of aurora formations using simultaneous measurements from in-situ data from FAST and surface observations from a THEMIS ground station. Data was taken when the FAST satellite passed through magnetic field lines conjugate to that of the ground stations. Conclusions from these events show the existence of distinct types of aurora that depend on longitude, latitude, and specific substorm.

As a part of the International Solar-Terrestrial Physics (ISTP) Program, the Polar spacecraft’s main mission was to collect images of the formation and development of the visible aurora over the poles [7]. The instruments in the Polar satellite collected magnetic field, electric field, and charged particle flux measurements over the poles [8]. Data from Polar has also been used to study alfvénic acceleration in the upper altitudes. Alfvén waves are the oscillatory motion of electrons and ions in the direction of the magnetic field. Janhunen *et al.* have identified a region at an altitude of 4-5 R_e (Earth radii) that is dominated by alfvénic acceleration of electrons [9]. This region along with another altitude region at 2-3 R_e , which is associated with electrostatic acceleration, contribute to auroral electron precipitation and energization [9].

The Viking and Freja satellite missions are both auroral research projects launched by the Swedish Space Corporation. These two satellites were some of the few projects that launched with the purpose of collecting magnetospheric and auroral measurements. With a lifetime of 15

months, the Viking satellite gathered electron acceleration data and magnetic and electric field data over the auroral ovals at high altitudes [10]. Using statistical data from Viking, Lindqvist [11] confirmed the parallel electric field over the poles and its relationship to altitude and plasma density. Measurements were collected in instances when the spin axis of the satellite lined up with the Earth's magnetic field. The Freja satellite followed the Viking project in 1992. One important characteristic of this satellite was the high spatial and temporal resolution measurements that were collected to perform detailed observations of narrow auroral bands [12]. Measurements from Freja have provided a more detailed look into sub-auroral electric fields [13] and electron acceleration in the poles [14].

The Time History of Events and Macroscale Interactions, or THEMIS, was a series of 5 small satellites, called probes, located inside the Earth's magnetosphere at different altitudes and orbits. Launched in 2007, the objective of the THEMIS mission was to survey the magnetotail region of the magnetosphere, see Figure 1.2. In particular, the series of satellites was tasked with “measuring the time sequences of substorm dynamics”[15].



**Figure 1.2: Schematic of THEMIS mission satellite layout, *THEMIS satellites*. 2014; Available from: http://www.nasa.gov/images/content/164101main_auroral_disruption_med.jpg.
Used under fair use, 2014**

Looking at early results from the THEMIS satellites, McFadden *et al.* found that cold plasma, plasma that is not directly in contact with the solar wind, may exist in the reconnection process

[15]. Recall from section 1.1.3, that the magnetic reconnection process is one of the driving factors for aurora formation. When the series of satellites are aligned towards the magnetotail, as shown in Figure 1.2, the reconnection event can be measured from magnetotail toward the Earth.

1.3 Motivation

The motivation for this research comes from the need to study the electromagnetic events, specifically the discrete and diffuse aurora, that occur at high latitudes over the Earth's geomagnetic poles. A number of various satellites have been placed to survey and study this auroral region and to investigate its relationship to the Sun's solar wind. However, in order to study the evolution of the electromagnetic activity around this region, it would be beneficial to have a two satellite mission capable of collecting simultaneous data at different altitudes over the auroral ovals. This thesis uses the Particle Swarm Optimization method to find an optimum two satellite configuration to view auroral events over the geomagnetic poles. In particular, the purpose is to research a series of instantaneous thrusts for two satellites that will maximize a series of event times in which both satellites are over the auroral region at different altitudes.

1.4 Research Approach

The solution to this problem is defined as an optimal series of instantaneous burns in order to maximize a set of objectives. A cost function is defined to maximize the total event time and minimize the total fuel cost used by both satellites. In addition, the possibility of minimizing the angle between the satellites' radius vectors is explored. This additional objective is given from the possibility of having both satellites above the same location over the Earth's surface but at different altitudes. Three scenarios are observed in this thesis: a single burn case, a double burn case, and a four burn case. In each of these cases, the Particle Swarm Optimization method is

used to identify the magnitude and direction of an optimum series of instantaneous burns. Finally, a comparison of the optimal solutions are made for all cases using two different orbital dynamics scenarios: a two-body problem with no perturbations and a two-body problem with added equatorial bulge effects, or the J_2 effect.

1.5 Thesis Outline

This paper is divided into 5 sections that will discuss the optimization problem given in the previous section and other related topics.

In Chapter 1, a literary review of related topics on space weather is given. Topics include the Sun's solar wind, the Earth's magnetosphere, and aurora formation in the lower atmosphere. Also included is a brief description of the satellite missions deployed to take measurements inside the Earth's magnetosphere.

A short introduction into the orbital mechanics used in this thesis is given in Chapter 2. This covers both the unperturbed and perturbed Keplerian orbits and the numerical solver used for satellite propagation.

A description of the Particle Swarm Optimization (PSO) method and the optimization problem is left for Chapter 3. This includes a brief description of past applications of PSO in astrodynamics and satellite mission planning. This chapter also gives the complete optimization problem outline and the selection of optimization parameters.

Finally, Chapters 4 and 5 give the results and conclusions based on the optimal solutions given by the optimization problem.

Chapter 2

Orbital Dynamics

This thesis will focus on the orbit of two satellites in Low Earth Orbit, LEO. The orbital analysis will involve simple two-body orbital mechanics, between a satellite and the Earth, and any added perturbations. Each satellite will be propagated independently of the other and there will be no coupling in the dynamics. The spacecraft motion is defined in the following sections.

2.1 Frame of Reference

The frame of reference chosen is the Earth Centric Inertial, or ECI. This coordinate system places the $\vec{\mathbf{I}}$ axis in the direction of the Vernal Equinox from some origin \mathbf{O} . The $\vec{\mathbf{I}}$ and $\vec{\mathbf{J}}$ axes are in the Earth's equatorial plane with the $\vec{\mathbf{K}}$ axis pointing in the direction of the Earth's spin axis, see Figure 2.1. This inertial reference frame is used in the derivation of the trajectory equation for the two-body problem. In Figure 2.1, the latitude (ϕ) and longitude (λ) are defined with respect to the ECI reference frame. The longitude is measured positively from the $\vec{\mathbf{I}}$ axis in the direction of the $\vec{\mathbf{J}}$ axis, while the latitude is measured positively from the equatorial plane in the direction of the $\vec{\mathbf{K}}$ axis. Normally, λ is measured from the Greenwich Meridian, a location near

the UK, and would rotate with the Earth. For simplicity, the coordinate system and the location of the Greenwich Meridian are stationary and the Greenwich Meridian is placed along the $\vec{\mathbf{I}}$ axis.

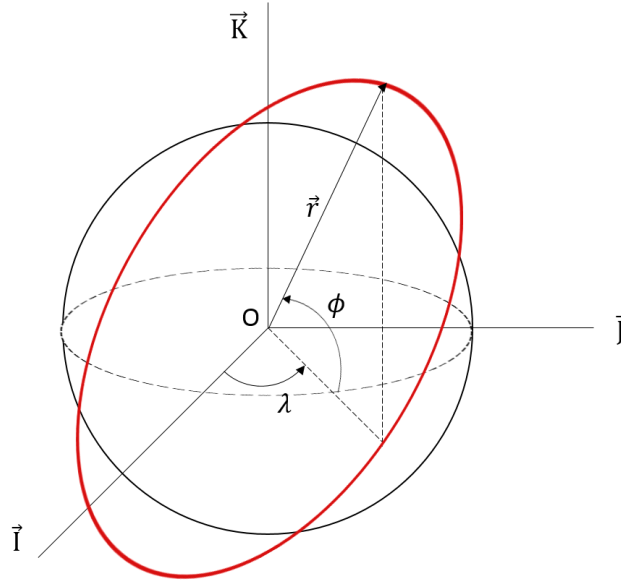


Figure 2.1: Earth Centered Inertial, ECI, reference frame. The latitude and longitude are defined in this figure, where the motion of a satellite is given in red.

2.2 Keplerian Two Body Problem

The orbital mechanics used in this paper is restricted to the two-body problem. The derivation is dependent on the following assumptions [17].

- Both bodies are perfect spheres and have uniform densities.
- The mass of the satellite is significantly less than the mass of the Earth.
- Coordinate system is inertial.

The derivation then uses Newton's Second Law and Newton's Law of Gravitational Attraction, see Eq. (2.1-2.2) and Figure 2.1.

$$\Sigma \vec{F} = \frac{d(m\vec{v})}{dt} = m\vec{a} \quad (2.1)$$

$$\vec{F}_g = -\frac{Gm_{\oplus}m_{sat}}{r^2}\left(\frac{\vec{r}}{r}\right) \quad (2.2)$$

The two-body equation is written as,

$$\ddot{\vec{r}} = -\frac{G(m_{sat} + m_{\oplus})}{r^2}\left(\frac{\vec{r}}{r}\right) \quad (2.3)$$

where G is the gravitational constant with a value of $6.6738 \times 10^{-11} \text{ km}^3 \text{ kg}^{-1} \text{ s}^{-2}$, m_{sat} is the mass of the satellite, m_{\oplus} is the mass of the Earth, and \vec{r} is the radius vector from the Earth to the satellite. Now the mass of the satellite is assumed negligible compared to the mass of the Earth, then the two-body nonlinear vector differential equation can be written as [17],

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} \quad (2.4)$$

Eq. (2.4) is expressed in Cartesian coordinates with μ as the gravitational parameter, defined as $\mu = Gm_{\oplus}$ with a value of $3.986 \times 10^5 \text{ km}^3/\text{s}^2$.

In reality, the shape of the Earth is not a perfect sphere, but this is a common assumption to use as an initial estimate. The solution to Eq. (2.4) is an important result in astrodynamics and a basis to understanding orbital mechanics. Two important properties can be derived from Eq. (2.4): conservation of specific angular momentum and conservation of specific mechanical energy.

The conservation of the angular momentum can be proved by first applying the cross product of \vec{r} with Eq. (2.4),

$$\vec{r} \times \ddot{\vec{r}} + \vec{r} \times \frac{\mu}{r^3}\vec{r} = 0 \quad (2.5)$$

Then recognize that the second term vanishes because $\vec{r} \times \vec{r} = 0$. The following definition can be used [17],

$$\frac{d}{dt}(\vec{r} \times \dot{\vec{r}}) = \dot{\vec{r}} \times \dot{\vec{r}} + \vec{r} \times \ddot{\vec{r}} = \vec{r} \times \ddot{\vec{r}} \quad (2.6)$$

The definition of the specific angular momentum is $\vec{h} = \vec{r} \times \dot{\vec{r}}$, therefore the derivative of the specific angular momentum is zero and this property is constant.

The conservation of the specific mechanical specific energy can be derived in a similar fashion. Start by applying the dot product of \vec{v} with Eq. (2.4),

$$\vec{v} \cdot \ddot{\vec{r}} + \vec{v} \cdot \frac{\mu}{r^3} \vec{r} = 0 \quad (2.7)$$

This can be simplified by using the definition of the dot product,

$$v\dot{v} + \frac{\mu}{r^3} r\dot{r} = 0 \quad (2.8)$$

Now the following derivative definitions can be used,

$$\frac{d}{dt} \left(\frac{v^2}{2} \right) = v\dot{v} \quad (2.9)$$

$$\frac{d}{dt} \left(-\frac{\mu}{r} \right) = \frac{\mu}{r^2} \dot{r} \quad (2.10)$$

Then the derivative of the specific mechanical energy can be written as:

$$\frac{d}{dt}(\xi) = \frac{d}{dt} \left(\frac{v^2}{2} - \frac{\mu}{r} \right) = 0 \quad (2.11)$$

This confirms that the specific mechanical energy, ξ , is conserved along the orbit.

Using the conservation of specific mechanical energy, the solution to Eq. (2.4) can be written in the following form:

$$r = \frac{p}{1 + e \cos(\nu)} \quad (2.12)$$

where r is the magnitude of the radius vector, p is the semi-parameter, e is the eccentricity and ν is the true anomaly. Eq. (2.12) is called the Trajectory Equation. It gives the magnitude of the

radius vector as the satellite moves around the body. Eq. (2.12) is an extension of Kepler's 1st Law and implies that satellite motion is not restricted to an ellipse, but may include all conic shapes [17]. A satellite's orbital motion can also be described using a set of orbital elements, see Table 2.1 and Figure 2.2.

Table 2.1: Keplerian orbital elements.

Semi-major axis, a	Related to conic sections. Measured as half of the line length between perigee and apogee for ellipses.
Eccentricity, e	Shape of the orbit (ellipse, parabola, or hyperbola). Related to conic sections.
Inclination, i	Tilt of the orbit plane. Measured from the \vec{I} - \vec{J} plane in the IJK coordinate system.
Right Ascension of the Ascending Node, Ω	Angle in the equatorial plane measured from the \vec{I} axis to the ascending node. The ascending node is the point where the satellite crosses from south of the equatorial plane to the north.
Argument of perigee, ω	Angle measured from Ω , points to location of perigee.
True anomaly, ν	Angle location of satellite with respect to perigee location.

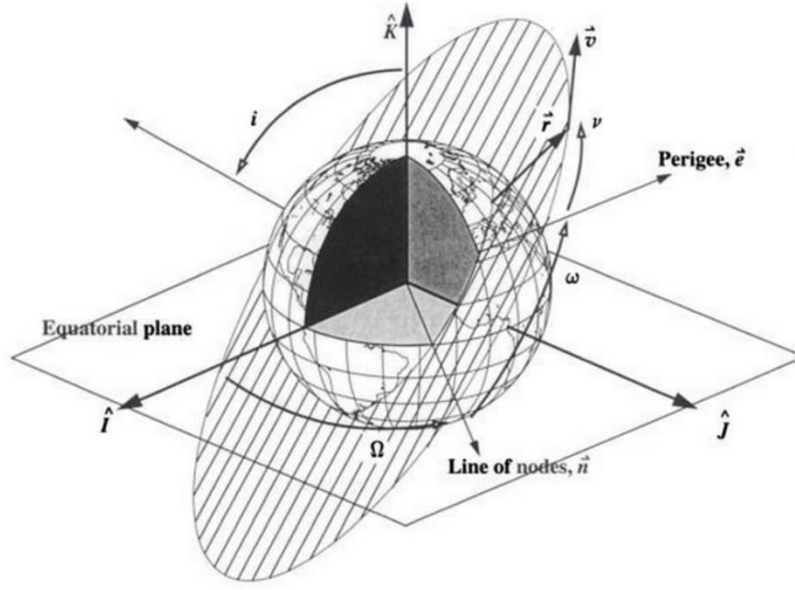


Figure 2.2: Classical orbital elements. Vallado, D.A. and W.D. McClain, *Fundamentals of Astrodynamics and Applications*. 2007: Springer. Used under fair use, 2014

2.3 J₂ Effect

The Earth is neither a perfect sphere nor does it have uniform density. In order to describe the perturbing motion of a non-spherical Earth, Vallado [17] gives the following aspherical potential function, based on gravitational potential energy,

$$U = \frac{\mu}{r} \left[1 + \sum_{l=2}^{\infty} \sum_{m=0}^l \left(\frac{R_{\oplus}}{r} \right)^l P_{l,m}[\sin(\phi_{gc_{sat}})] \{ C_{l,m} \cos(m\lambda_{sat}) + S_{l,m} \sin(m\lambda_{sat}) \} \right] \quad (2.13)$$

with the Earth's equatorial radius, R_{\oplus} , and the geocentric latitude and longitude represented as $\phi_{gc_{sat}}$ and λ_{sat} , respectively. The constants $C_{l,m}$ and $S_{l,m}$ represent spherical harmonics that are the basis for the mathematical modeling of the Earth's shape. l and m are defined as indices that correspond to the associated Legendre functions, $P_{l,m}$, defined as [17],

$$P_{l,m}[\gamma] = (1 - \gamma^2)^{\frac{m}{2}} \frac{d^m}{d\gamma^m} P_l[\gamma] \quad (2.14)$$

which uses the conventional Legendre polynomials defined as,

$$P_l[\gamma] = \frac{1}{2^l l!} \frac{d^l (\gamma^2 - 1)^l}{d\gamma^l} \quad (2.15)$$

Now, Eq. (2.13) can be separated by using the definition of zonal harmonics,

$$-C_{l,0} = J_l \quad (2.16)$$

Then the potential function takes the following form,

$$U = \frac{\mu}{r} \left[1 - \sum_{l=2}^{\infty} J_l \left(\frac{R_{\oplus}}{r} \right)^l P_l[\sin(\phi_{gcsat})] + \sum_{l=2}^{\infty} \sum_{m=1}^l \left(\frac{R_{\oplus}}{r} \right)^l P_{l,m}[\sin(\phi_{gcsat})] \{C_{l,m} \cos(m\lambda_{sat}) + S_{l,m} \sin(m\lambda_{sat})\} \right] \quad (2.17)$$

The only perturbation this thesis will include is that caused by the equatorial bulge of the Earth which is associated with the J_2 zonal harmonic. In order to write the perturbing accelerations of the J_2 effect in Cartesian coordinates only a portion of Eq. (2.17) is needed. The potential of this perturbation can be written as [17],

$$U_{J_2} = -\frac{3J_2\mu}{2r} \left(\frac{R_{\oplus}}{r} \right)^2 \left(\sin^2(\phi_{gc}) - \frac{1}{3} \right) \quad (2.18)$$

From Figure 2.1, the geocentric latitude ϕ_{gc} can be defined as $\frac{r_k}{r}$. The accelerations in Cartesian coordinates can be derived from the gradient of Eq. (2.18) along the different axes. First, a simplification of Eq. (2.18) is made,

$$U_{J_2} = -\frac{3J_2\mu}{2r} \left(\frac{R_{\oplus}}{r} \right)^2 \left(\frac{r_k^2}{r^2} - \frac{1}{3} \right) \quad (2.19)$$

Then, the perturbation accelerations can be derived as [17],

$$\vec{a}_{J_2I} = \frac{\partial U_{J_2}}{\partial r_I} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{R_\oplus}{r}\right)^2 \left[\left\{ 1 - 5 \left(\frac{r_K}{r}\right)^2 \right\} \frac{r_I}{r} \right] \quad (2.20)$$

$$\vec{a}_{J_2J} = \frac{\partial U_{J_2}}{\partial r_J} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{R_\oplus}{r}\right)^2 \left[\left\{ 1 - 5 \left(\frac{r_K}{r}\right)^2 \right\} \frac{r_J}{r} \right] \quad (2.21)$$

$$\vec{a}_{J_2K} = \frac{\partial U_{J_2}}{\partial r_K} = -\frac{3}{2}J_2 \left(\frac{\mu}{r^2}\right) \left(\frac{R_\oplus}{r}\right)^2 \left[\left\{ 3 - 5 \left(\frac{r_K}{r}\right)^2 \right\} \frac{r_K}{r} \right] \quad (2.22)$$

with J_2 as a non-dimensional constant of 1082.63×10^{-6} and the Earth's mean radius (R_\oplus) given a value of 6378 km. The new differential equation now takes the following form in the inertial IJK reference frame:

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \vec{a}_{J_2} \quad (2.23)$$

Equation (2.23) doesn't have an explicit solution, therefore an approximate orbit for each satellite will be found using a numerical solution, see next section. The J_2 effect is a significant perturbing force that causes secular variation of the Right Ascension of the Ascending Node, Ω , and the line of apsides, ω . The bulge near the equator plane “pulls” the satellite toward the Earth's equator, which causes a non-periodic mean motion of Ω [17]. This can be seen in Figure 2.3, where for positive inclinations, Ω moves toward the $\vec{\mathbf{I}}$ axis. Note that there is a point at 90° inclination where there is no regression. In the same manner, the J_2 effect causes a secular effect on the line of apsides, location of the orbit perigee. Figure 2.4 shows this effect and the inclinations, about 63° and 117° , where there is no regression [18].

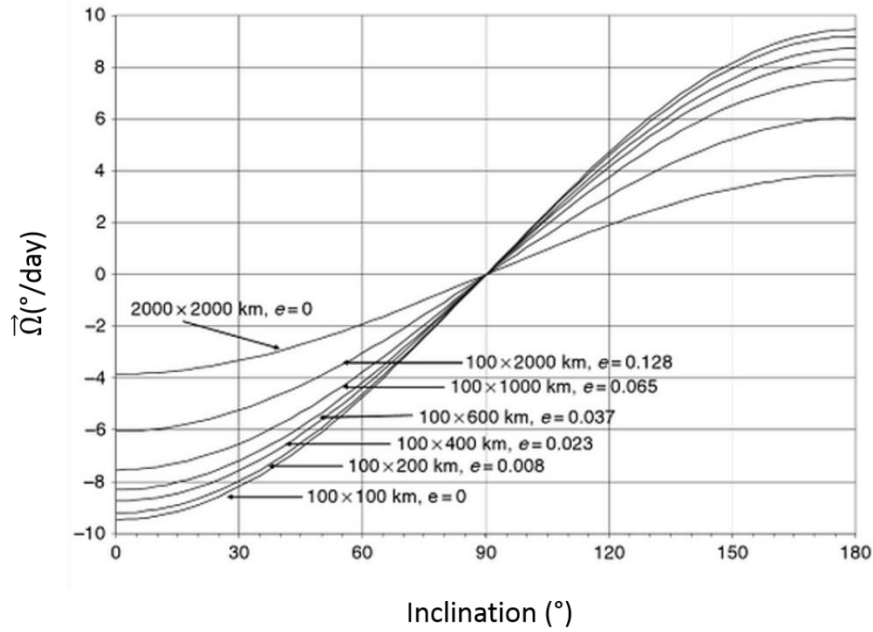


Figure 2.3: J_2 effect on the regression of the Right Ascension of the Ascending Node. Gurfil, P., *Modern Astrodynamics*. 2006: Elsevier Science. Used under fair use, 2014

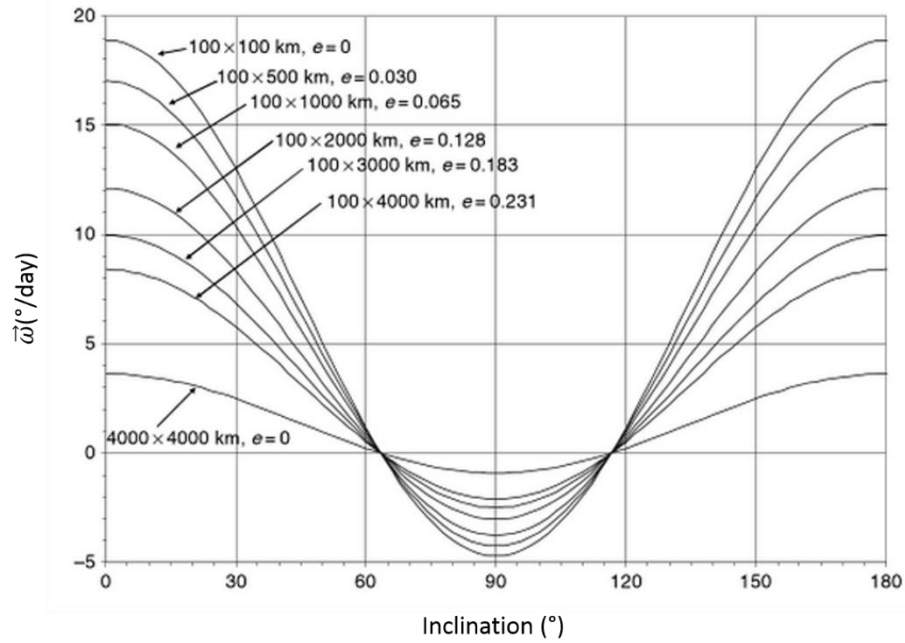


Figure 2.4: J_2 effect on the regression of the line of apsides. Gurfil, P., *Modern Astrodynamics*. 2006: Elsevier Science. Used under fair use, 2014

2.4 Numerical Integration

The numerical integration technique used is the *ode45* function, a built-in ordinary differential equation solver in MATLAB. This function uses a form of the Runge-Kutta single-step method

with a Taylor series expansion to extrapolate a solution to a differential equation [19]. The *ode45* function uses a Dorman-Prince 5(4) pair Runge-Kutta formulation. This method uses six evaluations for both a 4th and 5th order solution to the differential equation given. The difference between the two solutions is expressed as the local error of the lower order solution [20]. The numerical integration uses the state vector, defined as:

$$[X] = \begin{bmatrix} \vec{r} \\ \dot{\vec{r}} \end{bmatrix} = \begin{bmatrix} \vec{r} \\ \vec{v} \end{bmatrix} \quad (2.24)$$

with its first derivative,

$$[\dot{X}] = \begin{bmatrix} \vec{v} \\ -\frac{\mu}{r^3} \vec{r} + \vec{a}_{disturbance} \end{bmatrix} \quad (2.25)$$

where,

$$\vec{a}_{disturbance} = \vec{a}_{J_2} \quad (2.26)$$

2.5 Summary

This chapter described the orbital dynamics used in this thesis to describe the motion of the satellites. The two-body problem will be used to model the orbit of each satellite around the Earth. In this thesis, only elliptical orbits are considered. An unperturbed two-body model with and without the J_2 perturbation are considered to describe the motion of both satellites. The *ode45* built-in MATLAB function is used to obtain a time history of radii vectors for both satellites to determine the event times and average angles between the satellites used in the cost function, see next chapter. The optimal solutions obtained from the Particle Swarm Method will be compared using two different orbital mechanics models presented in this chapter.

Chapter 3

Particle Swarm Optimization

The Particle Swarm Optimization (PSO) method is part of a collection of computational intelligence algorithms. Developed by Eberhart and Kennedy [21], PSO is inspired by animal behavior such as birds flocking, fish schooling, and insects swarming. The PSO algorithm moves a population of “trial” solutions in order to find an optimum for a given equation or cost function. This has been shown to significantly cut down the number of iterations needed to find an optimum, but may be subject to convergence at local optima. In this chapter, first a brief introduction of the PSO formulation and two common forms used in research are provided. Followed by a description of the optimization problem outline.

3.1 Brief Introduction

The PSO is a population-based evolutionary computational method similar to other methods such as a Genetic Algorithm (GA) or Evolutionary Optimization. However, unlike these methods, the PSO algorithm retains a computational memory of all past solutions through the optimization process. For complex nonlinear problems, a population-based algorithm would be favorable to a single-solution algorithm because it could reduce the number of iterations and computation time. In a GA or Evolutionary Optimization method, trial solutions created for the next iteration are

formed from the fitness value (value of equation or cost function) of the current population through a process like mutation or crossover, see References [22, 23]. In these methods, “parent” solutions are chosen from the current iteration based on some selection function defined by the user. These “parent” solutions are used to form a new population by either combining elements of two “parents” (crossover) or changing an element of one “parent” (mutation). Note that the new population only relies on information from the current “parent” solutions and not from any past “parent” solutions.

A trial solution for the PSO algorithm may be written as:

$$J_i = f(x_i^1, x_i^2, \dots, x_i^D) \quad (3.1)$$

where D is the number of parameters optimized using the PSO method. For each trial solution, the PSO algorithm uses the parameters $x_i^{1,\dots,D}$ as the position coordinate for a particle in a D -dimensional hyperspace and classifies each population of particles as a swarm. Particles are given a position, velocity, and acceleration inside the hyperspace. In a one-dimensional space, the position of a particle is determined by the following equation [23]:

$$x_i(k+1) = x_i(k) + v_i(k) \quad (3.2)$$

where i is the number of particles in the swarm and k is the iteration. Alternatively for a D -dimensional problem, the position of a single particle in vector form may be expressed as:

$$\begin{bmatrix} x_i^1(k+1) \\ x_i^2(k+1) \\ \dots \\ x_i^D(k+1) \end{bmatrix} = \begin{bmatrix} x_i^1(k) \\ x_i^2(k) \\ \dots \\ x_i^D(k) \end{bmatrix} + \begin{bmatrix} v_i^1(k) \\ v_i^2(k) \\ \dots \\ v_i^D(k) \end{bmatrix} \quad (3.3)$$

The variable that impacts rate and accuracy of convergence in the PSO method is the velocity term, $v_i(k)$. The motion of the particle in the D -dimensional hyperspace is controlled by the particle velocity, which the user can manipulate to determine the how much of the hyperspace is explored and how quickly the swarm converges on an optimum. A higher velocity term would

allow a particle to search a broader region of the hyperspace, while a lower value would restrict its search region. Typically, the velocity is formulated by using the difference between the best and current particle positions. As a particle approaches an optimum, the velocity would decrease accordingly in order to close in on the optimum location.

Eberhart and Kennedy [24] describe two types of PSO models, a local and global version. The movement of the particles through the hyperspace is driven by the difference between the current position of the particle and some defined optimal position, usually a position that gives the best fitness value. The defined optimal position may be defined in two ways: the past best fitness position of the particle for all times and the past best fitness position from a group of particles for all times. Some references describe this as the cognitive and social aspects of the PSO, see Reference [25]. The cognitive aspect of a particle's velocity is the best position achieved by the particle up to the current iteration. The social aspect looks for the best position from either the entire swarm or a small neighborhood of particles. The difference between the local and global versions of PSO is in how the social portion of the velocity is calculated.

The local version of the PSO moves a particle according to the fitness of a small surrounding neighborhood of particles. This usually requires more iterations, and more computational effort, to converge depending on how wide the neighborhood is defined. However, because the particles are affected by an all-around best solution from a small surrounding region, it is less subject to converge at a local optimum. In the global version of PSO, the particles' motion are based on the best position of the entire swarm. This allows for a smaller number of iterations needed for convergence, which may also reduce computation time, but may increase the probability of converging at a local optimum.

This thesis will use the global version of PSO. The basic equation for the velocity and position for the global version is given as [23],

$$v_i(k+1) = v_i(k) + \gamma_{1i} \overbrace{(p_i - x_i(k))}^{\text{cognitive}} + \gamma_{2i} \overbrace{(G - x_i(k))}^{\text{social}} \quad (3.4)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (3.5)$$

with the personal best position of the i^{th} particle given as p_i and the global best position of the swarm given as G . The social and cognitive terms are multiplied by random numbers γ_{1i} and γ_{2i} , which are usually defined in the range of $0 \leq \gamma_{1i,2i} \leq 1$. The position vector is defined similar to Eq. (3.2). The velocity equation used in this thesis includes an additional inertia weight and a pair of acceleration constants added to Eq. (3.4). This is written as [23]:

$$v_i(k+1) = \psi(k)v_i(k) + A_1 \left[\gamma_{1i} \overbrace{(p_i - x_i(k))}^{\text{cognitive}} \right] + A_2 \left[\gamma_{2i} \overbrace{(G - x_i(k))}^{\text{social}} \right] \quad (3.6)$$

with a user defined inertia weight $\psi(k)$ and acceleration constants $A_{1,2}$. This is a more commonly used form of the global version of PSO, see [23]. The vector form of this equation for a D-dimensional problem can be written as:

$$\begin{aligned} \begin{bmatrix} v_i^1(k+1) \\ v_i^2(k+1) \\ \vdots \\ v_i^D(k+1) \end{bmatrix} &= \psi(k) \begin{bmatrix} v_i^1(k) \\ v_i^2(k) \\ \vdots \\ v_i^D(k) \end{bmatrix} + A_1 \begin{bmatrix} \gamma_{1i}^1 (p_i^1 - x_i^1(k)) \\ \gamma_{1i}^2 (p_i^2 - x_i^2(k)) \\ \vdots \\ \gamma_{1i}^D (p_i^D - x_i^D(k)) \end{bmatrix} \\ &+ A_2 \begin{bmatrix} \gamma_{2i}^1 (G^1 - x_i^1(k)) \\ \gamma_{2i}^2 (G^2 - x_i^2(k)) \\ \vdots \\ \gamma_{2i}^D (G^D - x_i^D(k)) \end{bmatrix} \end{aligned} \quad (3.7)$$

The acceleration parameters and the maximum velocity greatly influence the time to converge and the success of convergence. These will be explored in a later section.

The PSO global form with velocity defined in Eq. (3.6) is an attractive alternative to other population optimization techniques due to its simplicity and potential for low computational time. Pontani *et al.* [25] use a global version of the PSO method to solve constrained and unconstrained space trajectory problems. They explore periodic orbits in a restricted three-body problem and orbital transfers with finite and impulsive thrusts. An important verification made by Pontani was to confirm the optimality of the Hohmann transfer for orbital transfers using impulsive thrusts. Using an unconstrained formulation, Pontani *et al.* were able to verify that the Hohmann transfer is indeed a minimum energy transfer. Zhu *et al.* [26] used a hybrid PSO and differential evolution (DE) algorithm to maximize coverage time in a constellation of satellites. The DE algorithm is another population based method similar to the PSO. In his paper, Zhu suggests that the PSO and DE combination can be utilized to find an orbit design that would maximize coverage time and minimize fuel costs.

3.2 Optimization Problem

A cost function is formed to obtain an optimal orbit design to satisfy the following objectives:

- Maximize the total event time.
- Minimize the total fuel cost.
- Minimize the angle between the satellite radius vectors.

Comparison studies are performed to explore how the inclusion of the average angle objective affects the total event time and fuel cost in order to conclude its justification in the optimization cost function.

A series of instantaneous burns are applied to each satellite in an effort to find an optimal orbit design. This section will provide the outline for the optimization problem. First, the initial conditions of the satellites are given. Next, a description of the cost function formulation.

Finally, the subsequent MATLAB algorithm and the optimization parameters chosen are provided.

3.2.1 Initial conditions

The initial orbits of the satellites are chosen in order to allow each satellite to cross into the auroral oval four times per orbit. The satellites' orbital elements are given in Table 3.1 and Figure 3.1.

Table 3.1: Initial orbital elements of satellites.

	sat 1	sat 2
a	8206.24 km	8625.88 km
e	0.18	0.22
i	98°	98°
Ω	0°	0°
ω	90°	270°
v	0°	180°

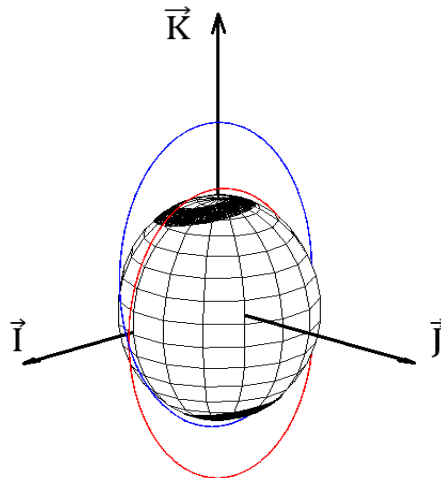


Figure 3.1: Initial orbit of satellites. Red: satellite 1. Blue: satellite 2.

The orbits are in the same plane and inclined in the direction of the geomagnetic pole, as seen in Figure 3.1. Initially, the satellites are placed over the geographic North Pole at different altitudes. In a situation where the eccentricities of both satellites were similar, the satellites would always line up over the geographic poles. For the unperturbed two-body problem, the motion of the satellites will be restricted to the same orbital plane.

The region of interest is a geomagnetic latitude range of $60^\circ \leq \phi_{gm} \leq 75^\circ$ in the northern hemisphere and $-60^\circ \leq \phi_{gm} \leq -75^\circ$ in the southern hemisphere, which will be referred to as the auroral oval, see shaded region in Figure 3.1. This is a loose definition of where the auroral oval is located, because the latitude region where auroral electron acceleration occurs may extend towards the equator, see Reference [27]. The difference between the geomagnetic poles and the geographic poles is estimated to be about 11° , see Section 1.1.2. An approximate location of the geomagnetic north pole is found in the International Geomagnetic Reference Field (IGRF) guide [28]. This source is provided by the International Association of Geomagnetism and Aeronomy (IAGA) as an approximation of the Earth's intricate magnetic field by using a complex mathematical model. The potential of the magnetic field is described as,

$$U_g(r, \theta_g, \phi_g, t) = \sum_{n=1}^N \sum_{m=0}^n \left(\frac{a_g}{r}\right)^{n+1} [g_n^m(t) \cos m\phi_g + h_n^m(t) \sin m\phi_g] \times P_n^m(\cos \theta_g) \quad (3.8)$$

where r is the radial distance from the center of the Earth, a_g is the magnetic reference spherical radius, θ_g is the geocentric co-latitude, ϕ_g is the east longitude, and $g_n^m(t)$ and $h_n^m(t)$ are the Gauss coefficients. Finally $P_n^m(\cos \theta_g)$ is a series of Legendre polynomials of n degree and of order m . Gauss coefficients for this potential model are obtained from in-situ data taken by various research organizations around the world. Using the IGRF, the latitude and longitude of the geomagnetic North Pole in 2014 is estimated to be at 80.2° N and 72.5° W, respectively.

Note that for simplicity, the $\vec{\mathbf{I}}$ axis is also the location of the Greenwich Meridian, the location from which to measure the longitude, see Figure 2.1. Also, the $\vec{\mathbf{I}}$ axis is the stationary location of the vernal equinox and it is not necessarily a unit vector that points to the Sun at all time.

3.2.2 Cost Function

The cost function is defined as:

$$J = -(1 - \alpha)C_v \frac{\Delta V_{tot}}{V_{esc}} - C_\theta \beta \frac{\theta_{ave}}{2\pi} + (\alpha)C_T \frac{T_{event}}{T_{day}} \quad (3.9)$$

where ΔV_{tot} is the total fuel cost of the satellites, θ_{ave} is the average angle between the satellite radius vectors, and T_{event} is the total event time. Weighting constants are applied to each term, defined as $C_v = 1$, $C_\theta = 10$, and $C_T = 1$, in order to bring the dimensionless terms within a certain range. Each term used in the cost function, Eq. (3.9), is further explored.

The total fuel cost is the summation of the magnitude of the instantaneous burn velocity.

This is given as:

$$\Delta V_{tot} = \sum_i^D |\Delta V_i| \quad (3.10)$$

where D represents the number of burns as well as the dimension of the optimization problem.

Note that all the minimizing terms in Eq. (3.9) are given a negative value. This method of representing minimizing terms with negative magnitudes has been shown to give decent performance in the optimization, in terms of convergence speed [25]. The summation is then divided by the Earth's escape velocity, V_{esc} , in order to create a dimensionless velocity term. The Earth's escape velocity is defined as the velocity needed in order to escape the Earth's gravitational pull, $V_{esc} = 11.2$ km/s. Each instantaneous burn is applied at the satellite's perigee in the direction of the satellite's velocity vector. The effect of the sign of the burn is shown in

Figure 3.2. A negative burn would decrease the eccentricity of the satellite and move the satellite's apogee closer to Earth, while a positive burn would increase the eccentricity and move the satellite's apogee away from Earth.

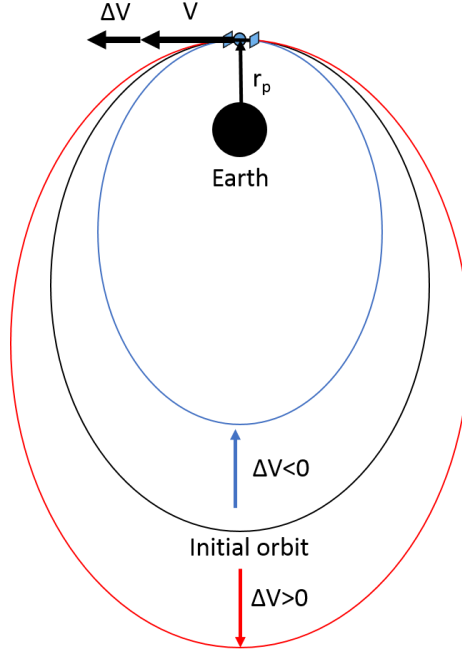


Figure 3.2: Effect of instantaneous burns.

The average angle between the radius vectors is calculated with the following function:

$$\theta_{ave} = \frac{\sum_i^N \int_{t_{a_i}}^{t_{b_i}} \theta(t) dt}{\sum_i^N t_{b_i} - t_{a_i}} \quad (3.11)$$

where t_a and t_b are the initial time and final time, respectively, for an event with a total of N number of events. An event is defined as the time range in which both satellites are inside the auroral oval. This may occur in the two latitude ranges previously discussed. The trapezoidal integration rule is used to determine the numerical integral of $\int_{t_{a_i}}^{t_{b_i}} \theta(t) dt$. The angle, θ , between the radius vectors of the satellites is found by using the definition of the dot and cross products:

$$\cos(\theta) = \frac{\vec{r}_1 \cdot \vec{r}_2}{\|\vec{r}_1\| \|\vec{r}_2\|} \quad (3.12)$$

$$\sin(\theta) = \frac{\|\vec{r}_1 \times \vec{r}_2\|}{\|\vec{r}_1\| \|\vec{r}_2\|} \quad (3.13)$$

To identify the proper quadrant for θ , Eq. (3.12-3.13) are combined to determine the tangent of the angle. Then the *atan2* function in MATLAB is used in the following equation:

$$\theta = \tan^{-1} \frac{\|\vec{r}_1 \times \vec{r}_2\|}{\vec{r}_1 \cdot \vec{r}_2} \quad (3.14)$$

Recognize that \vec{r}_1 and \vec{r}_2 are functions of time and are estimated by the propagation of the orbital dynamics of the two satellites. Finally, θ_{ave} is divided by 2π in order to create a dimensionless variable in the same manner as the velocity.

The total event time is calculated by using the radius vectors of both satellites from the Runge-Kutta numerical integration technique in the *ode45* function. Events are determined by looking at the geomagnetic latitudes of each satellite and determining when both satellites are inside one of the auroral latitude regions. First, the radius given in the IJK coordinate system, see Figure 2.1, is rotated into a geomagnetic coordinate system with the use of a 3-2 rotation. The rotation matrices are given as:

$$R_1(\delta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\delta) & \sin(\delta) \\ 0 & -\sin(\delta) & \cos(\delta) \end{bmatrix} \quad (3.15)$$

$$R_2(\delta) = \begin{bmatrix} \cos(\delta) & 0 & -\sin(\delta) \\ 0 & 1 & 0 \\ \sin(\delta) & 0 & \cos(\delta) \end{bmatrix} \quad (3.16)$$

$$R_3(\delta) = \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

$R_2(\delta)$ and $R_3(\delta)$ are defined as the rotation matrices for a rotation in the **J** and **K** axis, respectively. From [28], the location of the geomagnetic North Pole is at a geocentric latitude (φ_{geo}) and longitude (λ_{geo}) of 80.2° and -72.5° , respectively. The 3-2 rotation can be written as:

$$R_{rot} = \begin{bmatrix} \cos(\phi_{geo}) & 0 & -\sin(\phi_{geo}) \\ 0 & 1 & 0 \\ \sin(\phi_{geo}) & 0 & \cos(\phi_{geo}) \end{bmatrix} \begin{bmatrix} \cos(\lambda_{geo}) & \sin(\lambda_{geo}) & 0 \\ -\sin(\lambda_{geo}) & \cos(\lambda_{geo}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Now, the radius in the geomagnetic coordinate system (I'J'K') can be written as:

$$\vec{r}_{I'J'K'} = R_{rot} \vec{r}_{IJK} \quad (3.19)$$

Next, the geomagnetic latitude can be calculated:

$$\phi_{gm} = \tan^{-1} \left[\frac{r_{K'}}{\sqrt{r_{I'}^2 + r_{J'}^2}} \right] \quad (3.20)$$

The event times are determined by using the geomagnetic latitudes for satellites 1 and 2, ϕ_{gm}^1 and ϕ_{gm}^2 , respectively. A cubic-spline can be used to approximate the geomagnetic latitudes of each satellite and the time in which they cross into the desired latitude ranges. First, the latitudes when both satellites are inside the auroral oval can be found with the following functions:

$$f_1(t) = |\phi_{gm}(t)| - 60 \left(\frac{\pi}{180} \right) \quad (3.21)$$

$$f_2(t) = |\phi_{gm}(t)| - 75 \left(\frac{\pi}{180} \right) \quad (3.22)$$

There are three scenarios to observe during the satellites' propagation:

1. ϕ_{gm}^1 is inside the auroral region while ϕ_{gm}^2 is entering or exiting .
2. ϕ_{gm}^2 is inside the auroral region while ϕ_{gm}^1 is entering or exiting.
3. Both ϕ_{gm}^1 and ϕ_{gm}^2 are entering and/or exiting the auroral region.

For scenarios 1 and 2, the event time is associated with the satellite entering or exiting the auroral region. Scenario 2 is presented in Figure 3.3, where satellite 1 is displayed in red and satellite 2 is displayed in blue. The orbits are displayed in the new I'J'K' coordinate system and

the latitude regions are shown by dash lines. Figure 3.3 specifically shows the calculation of an initial event time, t_a . In scenario 3, the event time is given by the latest or earliest time of the satellite either entering or exiting. Figure 3.4 presents scenario 3 in the same coordinate system as Figure 3.3. In this scenario, both satellites cross into the auroral oval during the same time interval, therefore the initial event time, t_a , or final event time, t_b , for each satellite is calculated. The actual event time is then calculated from the following cases:

a) If satellites are entering:

$$t_a = \max(t_{a_{sat1}}, t_{a_{sat2}}) \quad (3.23)$$

b) If satellites are exiting:

$$t_b = \min(t_{b_{sat1}}, t_{b_{sat2}}) \quad (3.24)$$

c) If one satellite is entering while another is exiting. Calculate the time, t_c , for which each satellite crosses either inside or outside the auroral oval using Eq. (3.21-3.22). Then the event times can be calculated as:

$$t_a = \min(t_{c_{sat1}}, t_{c_{sat2}}) \quad (3.25)$$

$$t_b = \max(t_{c_{sat1}}, t_{c_{sat2}}) \quad (3.26)$$

Figure 3.4 displays case a of scenario 3.

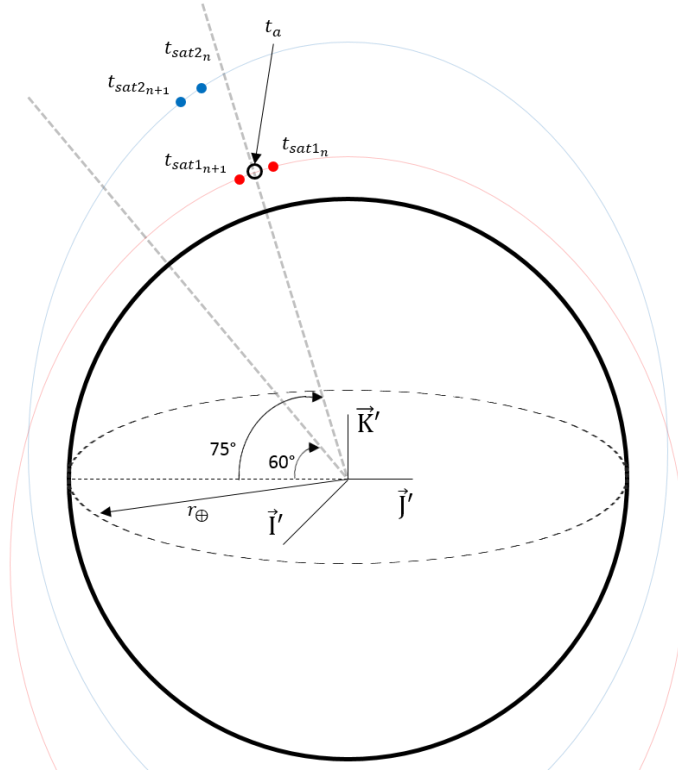


Figure 3.3: Scenario 2 – calculation of t_a is made with radius vectors from satellite 1.

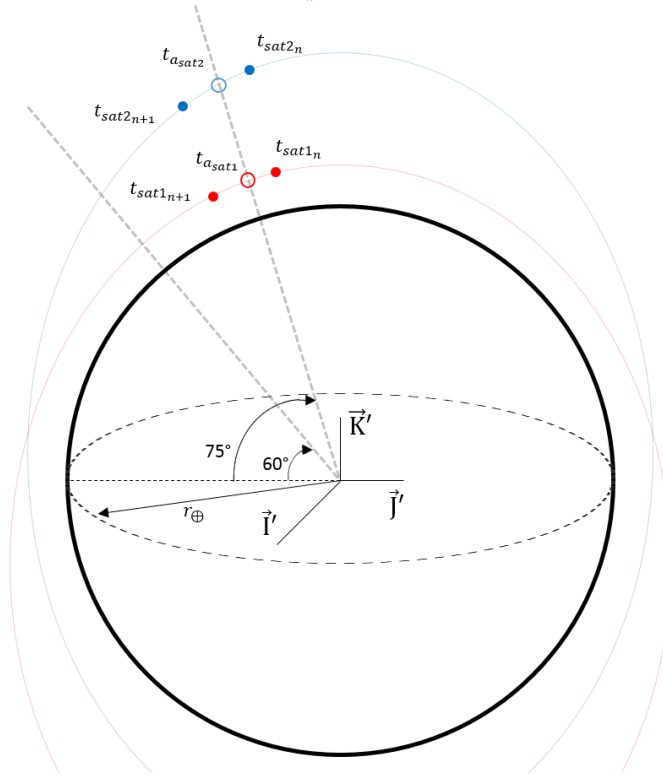


Figure 3.4: Scenario 3 - calculation of t_a is made with Eq. (3.23-3.26).

The event time can be approximated by using the geomagnetic latitude and its derivative with a cubic spline. The derivative of the latitude can be written as:

$$\frac{d}{dt}(\phi_{gm}) = \frac{1}{1 + \left(\frac{r_{K'}}{\sqrt{r_{I'}^2 + r_{J'}^2}} \right)^2} \left\{ \frac{v_{K'}}{\sqrt{r_{I'}^2 + r_{J'}^2}} + \frac{r_{K'}}{(r_{I'}^2 + r_{J'}^2)^{\frac{3}{2}}} [-r_{I'}v_{I'} - r_{J'}v_{J'}] \right\} \quad (3.27)$$

An approximate 3rd order equation can be made by using the latitude and its derivatives of the satellites. The cubic-spline equation is in the form of:

$$P_c(t) = a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0 \quad (3.28)$$

where the constants a_n are defined as [17]:

$$a_0 = f(t_n) \quad (3.29)$$

$$a_1 = \dot{f}(t_n)\Delta t \quad (3.30)$$

$$a_2 = -3f(t_n) - 2\dot{f}(t_n)\Delta t + 3f(t_{n+1}) - \dot{f}(t_{n+1})\Delta t \quad (3.31)$$

$$a_3 = 2f(t_n) + \dot{f}(t_n)\Delta t - 2f(t_{n+1}) + \dot{f}(t_{n+1})\Delta t \quad (3.32)$$

Note that Δt is defined as $t_{n+1} - t_n$ and that τ is defined in the interval $0 \leq \tau \leq 1$. Two latitude points are needed to find the exact time in which either satellite crosses into the auroral oval. Eq. (3.21-3.22) and their derivatives, which can be expressed with Eq. (3.27), can be used to obtain the cubic-spline function in Eq. (3.28). Then the possible event time can be found by using the roots of the cubic-spline function,

$$0 = a_3\tau^3 + a_2\tau^2 + a_1\tau + a_0 \quad (3.33)$$

The *roots* function in MATLAB is used to find the roots of Eq. (3.33). Only real roots, τ_r , are used to calculate the event times. The calculation of the possible event times t_a and t_b (associated with the entrance and exit, respectively, of the satellites into the auroral ovals) would then be defined as:

$$t_a, t_b = t_n + \tau_r \Delta t \quad (3.34)$$

For an N number of events, the total event time can be calculated as:

$$T_{event} = \sum_i^N t_{b_i} - t_{a_i} \quad (3.35)$$

In a similar manner to the other terms, the total event time is divided by one day, T_{day} expressed in seconds, in order to create a dimensionless variable.

Now, the evaluation of the cost function can be explored. The evaluation of the cost function is split into two phases: propagation and analysis. In the propagation phase, the motion of each satellite is calculated using the orbital mechanics model chosen and any impulsive thrusts to create an ephemeris, time history of position vector, for each satellite along the total propagation time. Then an analysis of the satellites positions, given by the ephemeris, determines the total event time and average angles, see Figure 3.5.

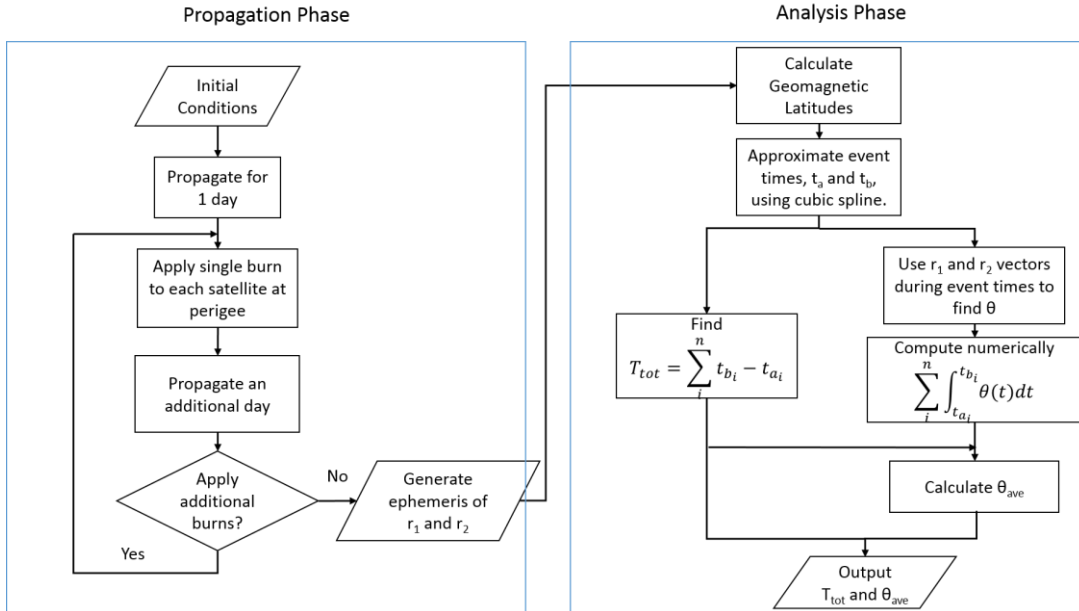


Figure 3.5: Calculate event times and average angle.

3.2.3 Theoretical Event Time and Angle

For the unperturbed case, a theoretical formulation of the total event time and average angle between the radius vectors can be derived from the trajectory equation and Kepler's Equation. This formulation is used in the single burn optimization results to verify the accuracy of the numerical simulations.

The formulation for the total event time uses the trajectory equation, Eq. (2.12). First, the radius vector is expressed in the PQW reference frame, see Figure 3.6. This frame places the axes \vec{P} and \vec{Q} in the orbital plane of the satellite and the \vec{Q} axis normal to the orbital plane. \vec{P} is in the direction of the perigee, \vec{W} is in the direction of the specific angular momentum, and \vec{Q} completes the triad [17].

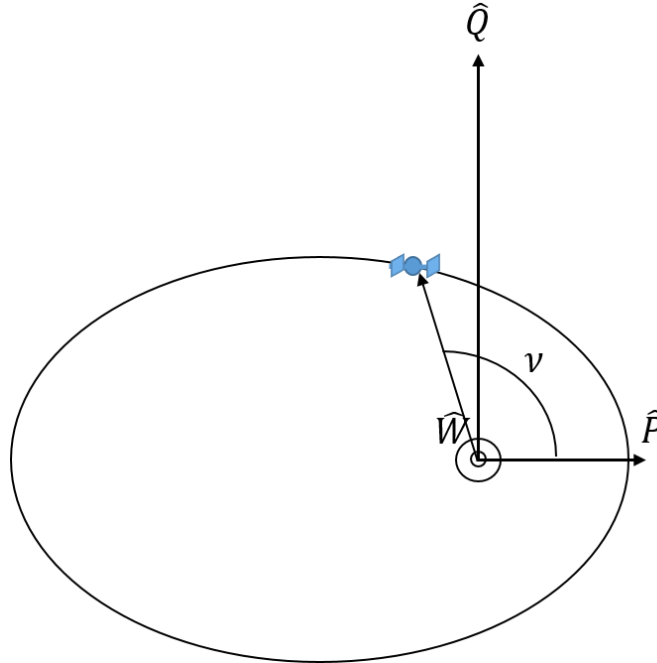


Figure 3.6: PQW Reference Frame

The radius vector is expressed as,

$$\vec{r}_{PQW} = r \cos(\nu) \hat{P} + r \sin(\nu) \hat{Q} + 0 \hat{W} \quad (3.36)$$

where the radius magnitude, r , may now be defined as,

$$r = \frac{a(1 - e^2)}{1 + e \cos(\nu)} \quad (3.37)$$

Now, Eq. (3.36) can be moved to the IJK reference frame by using a 3-1-3 rotation. The rotation matrices are expressed in Eq. (3.15-3.17) and the rotation matrix has the following form,

$$R_{IJK}^{PQW} = \begin{bmatrix} \cos(\omega) & \sin(\omega) & 0 \\ -\sin(\omega) & \cos(\omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(i) & \sin(i) \\ 0 & -\sin(i) & \cos(i) \end{bmatrix} \begin{bmatrix} \cos(\Omega) & \sin(\Omega) & 0 \\ -\sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.38)$$

This is the rotation matrix to move the reference frame from the IJK reference frame to the PQW frame. In order to reverse the rotation, the transpose of Eq. (3.38) is needed,

$$R_{PQW}^{IJK} = (R_{IJK}^{PQW})^T \quad (3.39)$$

Now the radius vector is written as,

$$\vec{r}_{IJK}(\nu) = R_{PQW}^{IJK} \vec{r}_{PQW}(\nu) \quad (3.40)$$

Note that the radius is a function of the true anomaly, ν , and not time. The components of the radius vector can be rotated to the geomagnetic coordinate system, I'J'K', with a 3-2 rotation, see Eq. (3.18):

$$\vec{r}_{I'J'K'}(\nu) = R_{rot} \vec{r}_{IJK}(\nu) \quad (3.41)$$

Next, the geomagnetic latitude can be calculated with,

$$\phi_{gm}(\nu) = \tan^{-1} \left[\frac{r_{K'}}{\sqrt{r_{I'}^2 + r_{J'}^2}} \right] \quad (3.42)$$

The following functions are used to find the true anomaly location where the satellites cross into the auroral oval,

$$f_1(\nu) = |\phi_{gm}(\nu)| - 60 \left(\frac{\pi}{180} \right) = 0 \quad (3.43)$$

$$f_2(v) = |\phi_{gm}(v)| - 75 \left(\frac{\pi}{180} \right) = 0 \quad (3.44)$$

The solution of Eq. (3.43-3.44) will give the true anomaly location, v_c . The f_{zero} function in MATLAB is used to find the zeros of Eq. (3.43-3.44). Now Kepler's Equation can be used to find the time associated with the true anomaly locations. A new orbital element is introduced, the eccentric anomaly. In the PQW reference frame, an auxiliary circle is drawn from the orbit ellipse, see Figure 3.7. The circle has a radius value equal to the semi-major axis of the orbit ellipse. In this thesis, only ellipses are considered so any other orbit shapes (hyperbolas or parabolas) are not included. At point B along the orbit, the true anomaly is defined as the angle between perigee and the radius vector, \vec{r} . The eccentric anomaly for this same point is defined as the angle at the center of the auxiliary circle measured from perigee of the orbit to point B' in the edge of the auxiliary circle. B' is formed by extending the line \overline{BC} to the edge of the auxiliary circle, where the line \overline{BC} is normal to a line connecting the origin, O, and the perigee, P, see Figure 3.7.

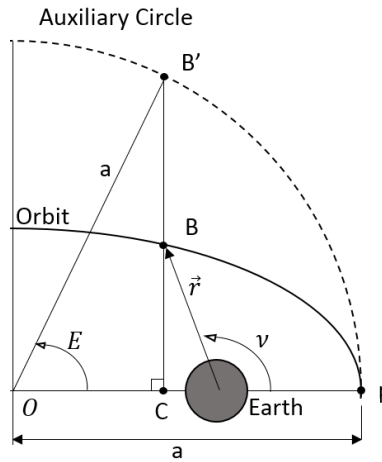


Figure 3.7: Eccentric Anomaly.

The eccentric anomaly is related to the true anomaly by [17],

$$\tan \frac{E}{2} = \sqrt{\frac{(1-e)}{(1+e)}} \tan \frac{\nu}{2} \quad (3.45)$$

The trajectory equation can now be written in terms of the eccentric anomaly,

$$r = a(1 - e \cos E) \quad (3.46)$$

The event times are found by using Kepler's Equation. Kepler's Equation relates the time of flight between two points along an orbit to their positions, more often expressed by using the eccentric anomaly. This can be derived by using a definition of the specific angular momentum,

$$h = r^2 \dot{\nu} \quad (3.47)$$

Both sides are integrated from a time T , where the satellite passes through the perigee, to a time t .

$$\int_T^t h dt = \int_0^\nu r^2 d\nu \quad (3.48)$$

This can be expressed with the eccentric anomaly, E , with the following expression from Vallado [17],

$$d\nu = \frac{a\sqrt{1-e^2}}{r} dE \quad (3.49)$$

Substituting Eq. (3.49) into Eq. (3.48) and simplifying,

$$h(t - T) = \int_0^E r^2 \frac{a\sqrt{1-e^2}}{r} dE \quad (3.50)$$

Recognize that the specific angular momentum is constant for the unperturbed two-body problem. By using Eq. (3.46) inside Eq. (3.50) and integrating,

$$h(t - T) = a^2 \sqrt{1-e^2} (E - e \sin E) \quad (3.51)$$

where specific angular momentum, h , can be defined with the mean motion, n :

$$h = na^2 \sqrt{1-e^2} \quad (3.52)$$

and,

$$n = \sqrt{\frac{a^3}{\mu}} \quad (3.53)$$

finally,

$$(t - T) = \sqrt{\frac{a^3}{\mu}} (E - e \sin E) \quad (3.54)$$

Eq. (3.54) is Kepler's Equation. Although a more general form can be written as [17]:

$$t - t_0 = \sqrt{\frac{a^3}{\mu}} [2k\pi + (E - e \sin(E)) - (E_0 - e \sin(E_0))] \quad (3.55)$$

which relates the change in time between any two positions along an orbit. In Eq. (3.55), k is the number of perigee passes between the two positions of interest. This form is only applicable to ellipses. The calculation of the event times, t_a and t_b , is completed by using Eq. (3.55) with the true anomalies, v_c , expressed as eccentric anomalies, E_c , with Eq. (3.45).

The average angle can be found by using the event times calculated with Kepler's Equation, Eq. (3.55), to approximate the integral in Eq. (3.11). The integral is approximated with the trapezoidal rule given a series of ten consecutive times between the event times, t_a and t_b . The angle between the two satellite vectors is given by,

$$\theta(v_1, v_2) = \tan^{-1} \frac{\|\vec{r}_1(v_1) \times \vec{r}_2(v_2)\|}{\vec{r}_1(v_1) \cdot \vec{r}_2(v_2)} \quad (3.56)$$

where \vec{r}_1 and \vec{r}_2 may be expressed in the I'J'K' coordinate system. As opposed to Eq. (3.14), the angle equation is now a function of the true anomaly of both satellites. This complicates the problem, because the satellites are not in the same orbit. The times between the event times, t_a and t_b , must be found by solving Kepler's Problem. This is in conjunction with Kepler's Equation, except now the final positions, E_f or v_f , are desired from the time of flight and initial

positions, E_0 or v_0 . This is one of the most troublesome problems in astrodynamics because it doesn't have a non-series closed form solution and requires an approximate solutions through numerical methods. In this research, the solution is approximated by using a Newton-Raphson method. The general form of the Newton-Raphson is [17],

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (3.57)$$

This equation is derived from a truncated Taylor series expansion of a function in the form of $f(y) = 0$. The Newton-Raphson uses the slope of the approximate answer to calculate the next guess. Kepler's problem can be expressed in the general form of the Newton-Raphson [17],

$$E_{n+1} = E_n + \frac{M - E_n + e \sin E_n}{1 - e \cos E_n} \quad (3.58)$$

Eq. (3.58) would be iterated until the following ending criterion is applied:

$$|E_{n+1} - E_n| < \epsilon \quad (3.59)$$

where ϵ is given a value of 1×10^{-10} . Carefully selecting a starting value of E_n affects the converging iterations. Vallado [17] proposes a number of starting E_n values for different orbits.

In this thesis, the following starting values are chosen based on the following conditions [17]:

- If $M > -\pi$ and $M < 0$ or if $M > \pi$. Use a starting value of $E_n = M - e$.
- Else, use a starting value of $E_n = M + e$.

with the mean anomaly, M , defined as:

$$M = (t - t_0) \sqrt{\frac{\mu}{a^3}} \quad (3.60)$$

For each time desired, the positions v_1 and v_2 must be calculated by solving Kepler's Problem for each positions. After the positions $v_{1,2}$ are found, then the angles may be found and the approximate integral can be evaluated. This same procedure is repeated for every event throughout the propagation.

3.3 PSO algorithm

The PSO algorithm used in this thesis is taken from a previously coded MATLAB function, see Reference [23]. This function uses the global form of the PSO algorithm previously described.

The following is an outline of this algorithm.

First, the initial swarm (population) is generated as a collection of particles with dimension D whose position and velocity are stochastically selected. The position of an i^{th} particle can be written as,

$$x_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \dots \\ x_i^D \end{bmatrix} \quad (3.61)$$

and the velocity as,

$$v_i = \begin{bmatrix} v_i^1 \\ v_i^2 \\ \dots \\ v_i^D \end{bmatrix} \quad (3.62)$$

Next, the initial swarm is placed in an iterative process that changes the position and velocity of the particles. The k iteration of the process has the following steps,

1. For $i = 1 \dots P$ (where P is the number of particles)
 - a. Evaluate the cost value, $J_i(k)$, for each particle i .
 - b. Determine the local best position, p , of each particle i .

$$p_i(k) = x_i(l) \quad (3.63)$$

where [25],

$$l = \arg \left(\max_{d=1, \dots, k} J_i(d) \right) \quad (3.64)$$

2. Determine the global best position, G , of the swarm for all time.

$$G(k) = p_q(k) \quad (3.65)$$

with the following definitions [25],

$$q = \arg \left(\max_{i=1 \dots P} F_i(k) \right) \quad (3.66)$$

$$F_i(k) = \max_{d=1, \dots, k} J_i(d) \quad (3.67)$$

3. Update the new velocities according to Eq. (3.6),

$$v_i(k+1) = \psi(k)v_i(k) + A_1[\gamma_{1i}(p_i(k) - x_i(k))] + A_2[\gamma_{2i}(G(k) - x_i(k))] \quad (3.68)$$

with user defined inertia weight $\psi(k)$, acceleration constants $A_{1,2}$, and random variables

$\gamma_{1i,2i}$. Determine if $v_i(k+1)$ is within the bounded particle velocity range defined

as $v_{min} \leq v_i(k+1) \leq v_{max}$. If $v_i(k+1)$ is not within the range imposed, then the

following apply,

a) If $v_i(k+1) < v_{min}$, then $v_i(k+1) = v_{min}$.

b) If $v_i(k+1) > v_{max}$, then $v_i(k+1) = v_{max}$.

4. Update the position vector for particle $x_i(k+1)$ using Eq. (3.2). Limit the position of a

particle to a user specified range: $x_{min} \leq x_i(k+1) \leq x_{max}$. If $x_i(k+1)$ is outside of

range, apply the following:

a) If $x_i(k+1) < x_{min}$, then $x_i(k+1) = x_{min}$.

b) If $x_i(k+1) > x_{max}$, then $x_i(k+1) = x_{max}$.

After K iterations, the optimal position, $G(K)$, and optimal cost value are given by Eq. (3.65-

3.67). As previously mentioned, the formulation of the velocity vector will affect a number of

properties in the optimization process, such as computational speed and accuracy of

convergence. These properties are also dependent on the user defined constants: the inertia

weight $\psi(k)$ and the acceleration constants $A_{1,2}$. The inertia weight is the constant value that

dictates the influence of the previous velocity for the current iteration. Shi *et al.* [29] have stated

that an inertia weight in the range of 0.9-1.2 is sufficient to obtain reasonable global optimum convergence and lower iterations. However, they also recommend that a linear decreasing inertia weight may also be favorable for better performance. An inertia weight linearly decreasing from 0.9→0.4 has been shown to give better performance, in terms of number of iterations, than using a constant inertia weight of 0.9 or 0.4. After a certain number of iterations, the particle will start to converge towards an optimum and a decreasing inertia weight would allow faster and more accurate convergence. The PSO algorithm used in this thesis will include a linearly decreasing inertia weight with a range of $0.4 \leq \psi(k) \leq 0.9$.

The acceleration constant $A_{1,2}$ impacts the influence of the social and cognitive portions of the velocity calculation, see Eq. (3.6). A higher number would allow the particle to search a bigger region of the hyperspace throughout the optimization. Although it is not really suggested that the search region be expanded by using these constants. The maximum velocity is used most often to control the search region of the swarm in the D-dimension hyperspace. The absolute maximum velocity in this thesis is defined as,

$$v_m = |v_{min}| = |v_{max}| \quad (3.69)$$

where now the velocity $v_i(k + 1)$ is bounded by $-v_m \leq v_i(k + 1) \leq v_m$. A bigger v_m will allow the particle to search a broader region of the hyperspace. Shi [29] suggests that a v_m value equal to x_{max} gives decent performance results, but this result is only applicable to the Schaffer function used in his research. Ultimately, the choice of these parameters are problem dependent and found after certain trial-error. The PSO parameters used in the optimization are given below.

Table 3.2: Particle Swarm Optimization constants

PSO Parameters	
v_m	$0.5 x_{\max}$
A_1	1
A_2	1
K	500
K_c	450
P	40

with K as the maximum number of iterations, P as the number of particles, and K_c as the iteration when the inertia weight will permanently change to a constant value of 0.4.

In this thesis, the positions of the particles will be the instantaneous burns applied to the satellites, ΔV . A sample run for the case with 2 instantaneous burns and with $\alpha=0.8$ and $\beta=0.5$ is shown in Figure 3.8. In this run, the cost function was in the form of Eq. (3.9) with the PSO parameters displayed in Table 3.2. The motion of the satellites is given by the unperturbed two-body problem. After 300 iterations, the particles begin to converge towards a specific optimum. Although, there are a couple of particles that seem to oscillate between the optimum solution. This can be attributed to the inertia weight chosen and the acceleration constants for the social and cognitive aspect of the velocity. The oscillating particle may be located at a position where the difference between its current position and its local best and global best are close in value. This would explain why it seems to be “stuck” near a certain value.

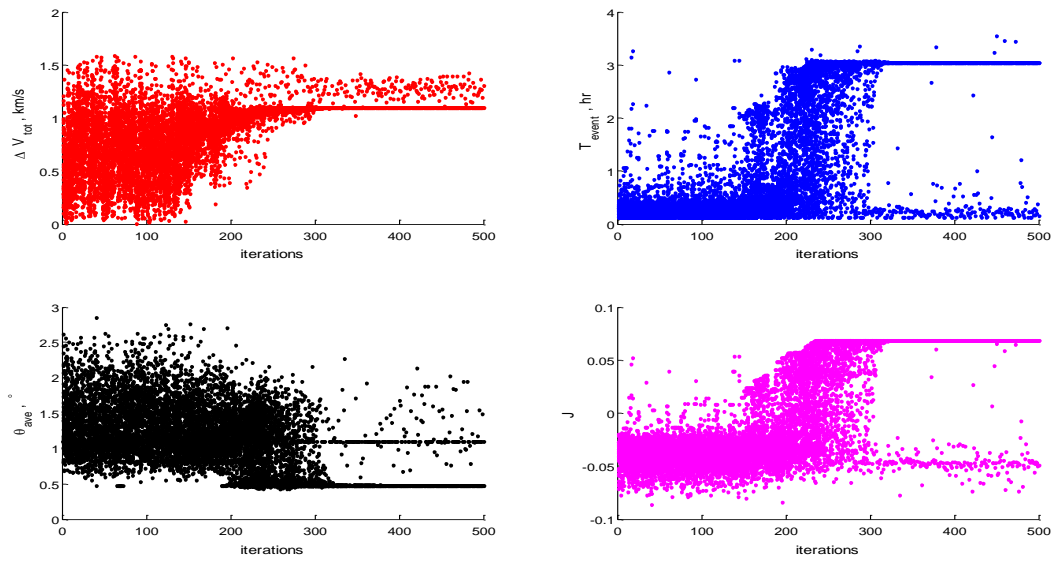


Figure 3.8: Cost terms for case of $\alpha=0.8$ and $\beta=0.5$.

Chapter 4

Results and Discussion

In this chapter, the optimization results are given for three cases: a single burn is applied only to one satellite, a single burn is applied to both satellites, and two burns are applied to each satellite. First, the optimization is analyzed by using a standard two-body problem with no perturbations. Next, a perturbed motion due to the equatorial bulge of the Earth, the J_2 effect, is used to express the satellites' motion.

4.1 Unperturbed Two-Body Motion

Optimal burn velocities are found for a system of two satellites orbiting the Earth whose motion is determined by an unperturbed two-body problem. Three cases are represented: a single burn case, a double burn case, and a four burn case. The optimal solutions for the three cases are generated from the PSO algorithm described in Section 3.3.

4.1.1 Single Burn Optimization

Two scenarios are presented: a single burn is only applied to satellite 1 and a single burn is only applied to satellite 2. In each scenario, the satellite remains in its initial orbit if no burn is

applied. Recall that all burns are applied at the perigee of each satellite's orbit after some propagation period and at the direction of the velocity vector.

The optimal configuration for these two scenarios is found by using the following cost function, which is a special case of Eq. (3.9) without the average angle included in the optimization ($\beta = 0$),

$$J = -(1 - \alpha)C_v \frac{\Delta V_{tot}}{v_{esc}} + \alpha C_T \frac{T_{event}}{T_{day}} \quad (4.1)$$

with the constants C_v and C_T previously defined in Section 3.2.2 and the variable α defined as $\alpha = \{0, 0.1, \dots, 1\}$. In order to better understand the physics of the single burn case using the varying α values in Eq. (4.1), a parameter study is provided in Figures 4.1-4.2. This is also useful in verifying the PSO results provided later in this section.

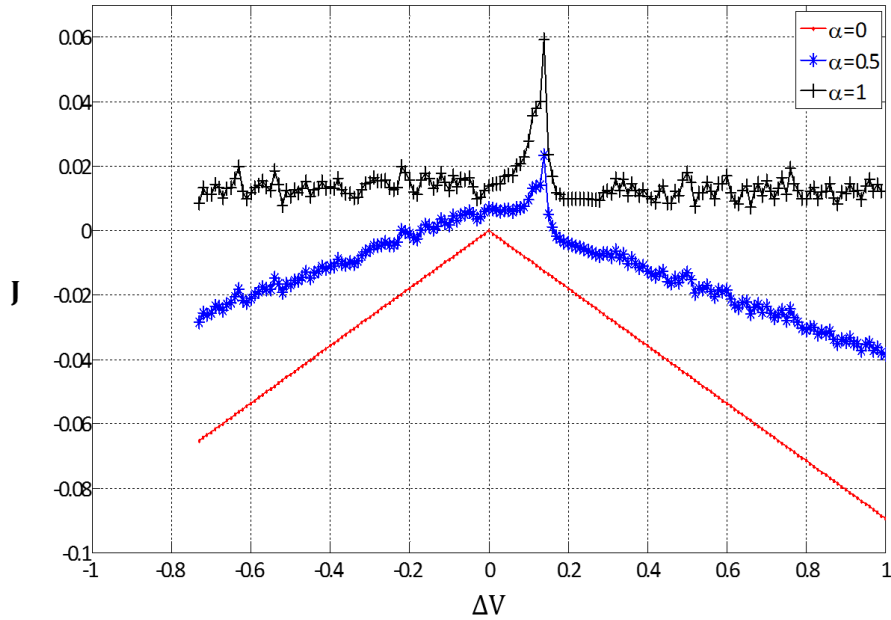


Figure 4.1: Satellite 1 single burn cost.

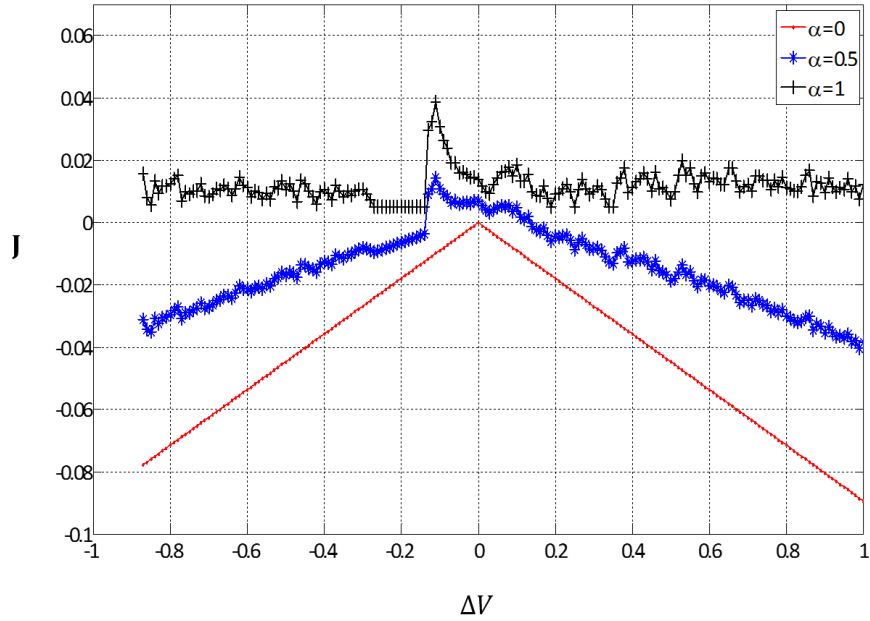


Figure 4.2: Satellite 2 single burn cost.

Notice that the range of applicable burn velocities differ in the two scenarios. This is an effect of the initial orbit chosen, which limits the deceleration burn ($-\Delta V$) that can be applied to each satellite before it crashes into the Earth's surface. Satellite 1 has an applicable burn velocity range of $-0.73 \leq \Delta V \leq 1$ while satellite 2 has a range of $-0.86 \leq \Delta V \leq 1$. The α value can be described as an additional weight applied to the cost function that affects the total event time and the fuel cost. Increasing α , increases the impact of the total event time in the optimization while also decreasing the impact of the fuel cost. This explains why the shape of the cost function changes with an increasing α . In addition, the total event time in the cost function (when $\alpha \neq 0$) creates an oscillatory motion in the cost value, see Figures 4.3-4.4.

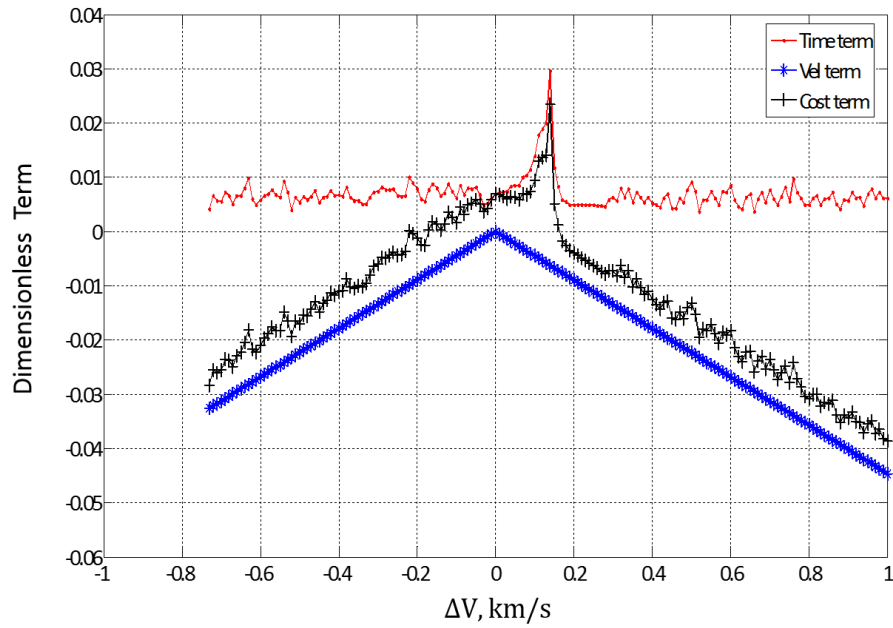


Figure 4.3: Cost function terms for satellite 1 single burn at $\alpha=0.5$.

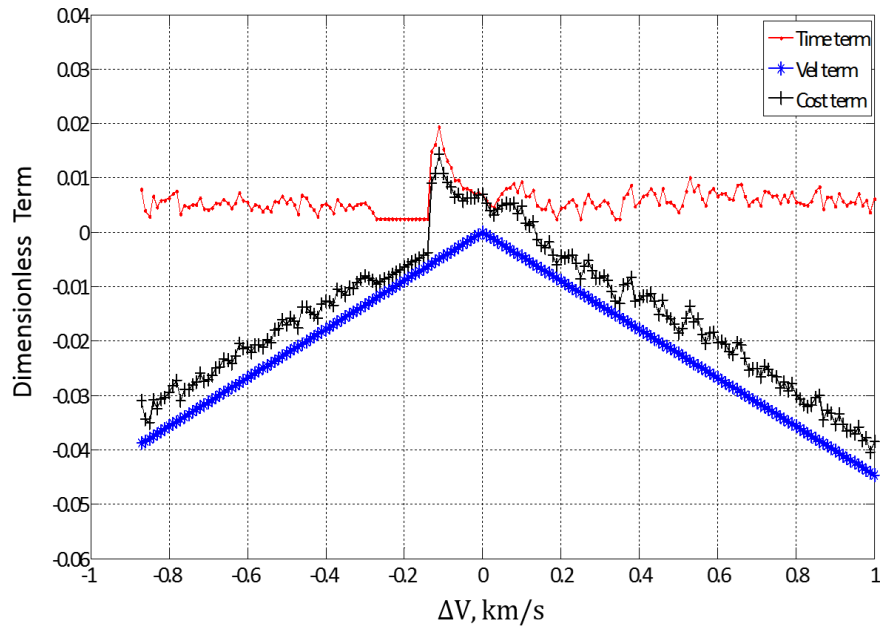


Figure 4.4: Cost function terms for satellite 2 single burn at $\alpha=0.5$.

A comparison study is completed between the formulation of the cost function using a numerical ODE solver and the trajectory equation for two-body motion. This study is completed to verify the accuracy of numerical ODE solver with respect to the trajectory equation. Sections 3.2.2-

3.2.3 describe the formulation of the cost function using a numerical ODE solver and the trajectory equation. Figure 4.5 shows the dimensionless time term using both simulations.

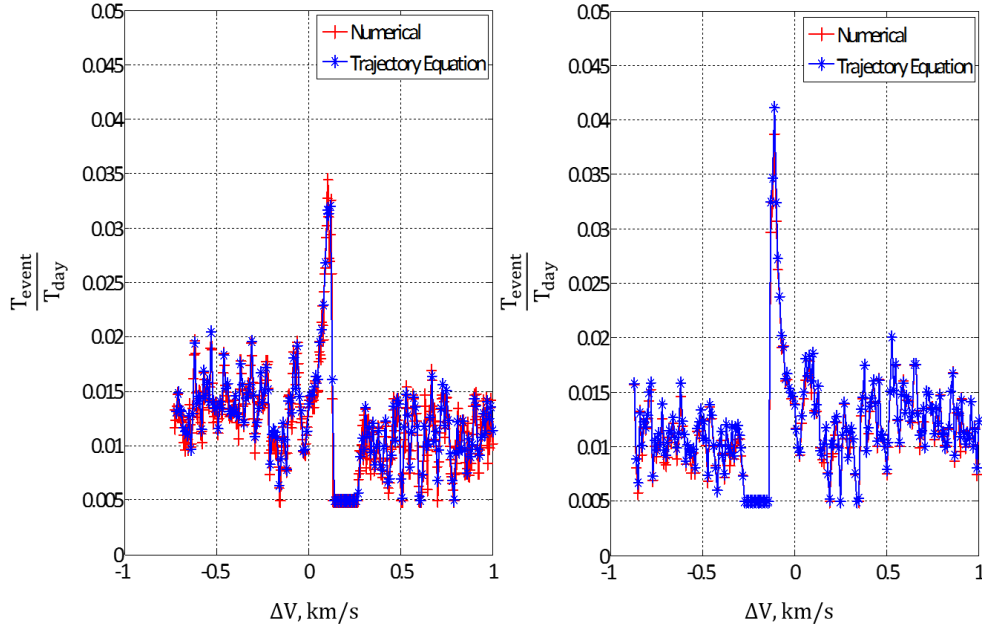


Figure 4.5: Single burn optimization time. Left: satellite 1 burn only. Right: satellite 2 burn only.

The numerical and theoretical formulations exhibit the same behavior in terms of oscillations in total event time calculation for both satellites. The difference in agreement between a satellite 1 burn and a satellite 2 burn is caused by the method used to calculate the total event time in the numerical and theoretical calculations. In the numerical case, the event times are calculated using radius vectors approximated by a runge-kutta technique and a cubic-spline, while these same events are approximated by solving Eq. (3.43-3.44) in the theoretical formulation. Next, a comparison of the total event time and the number of events is given in Figure 4.6.

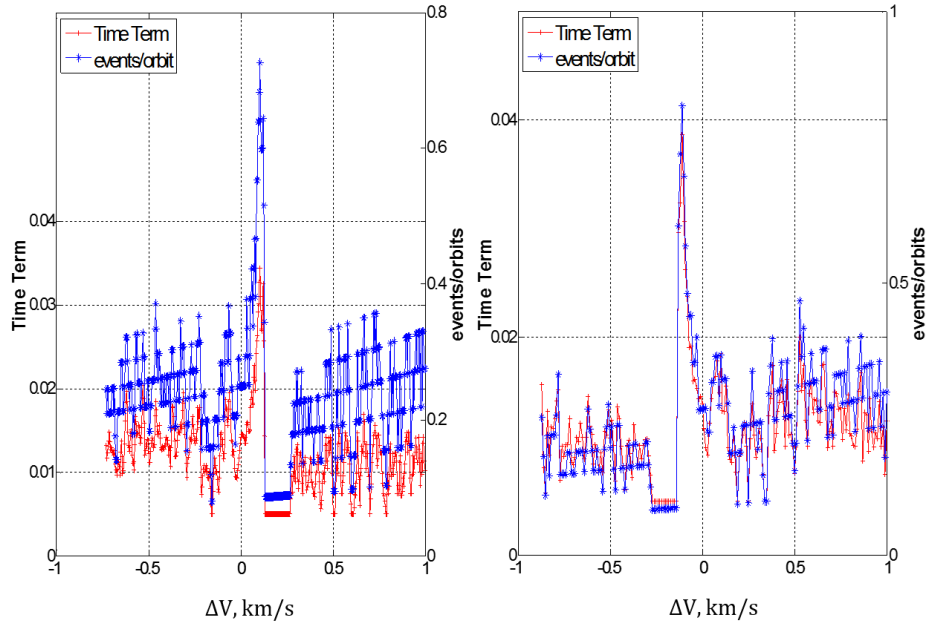


Figure 4.6: Events and Time. Left: satellite 1 burn only. Right: satellite 2 burn only.

The number of events oscillates in the same manner as the total event time seen in Figure 4.6.

Therefore a small change of burn velocity will cause the total number of events to change which will affect the total event time calculated. Note that there is a region directly following the location of maximum T_{event} where both the number of events and time remain constant. This region of ΔV is associated with an orbit configuration in which both satellites never see each other over the auroral oval after the instantaneous burns are applied.

Using the PSO algorithm described in Section 3.3, a plot of the optimal solutions with the varying α can be represented in Figure 4.7.

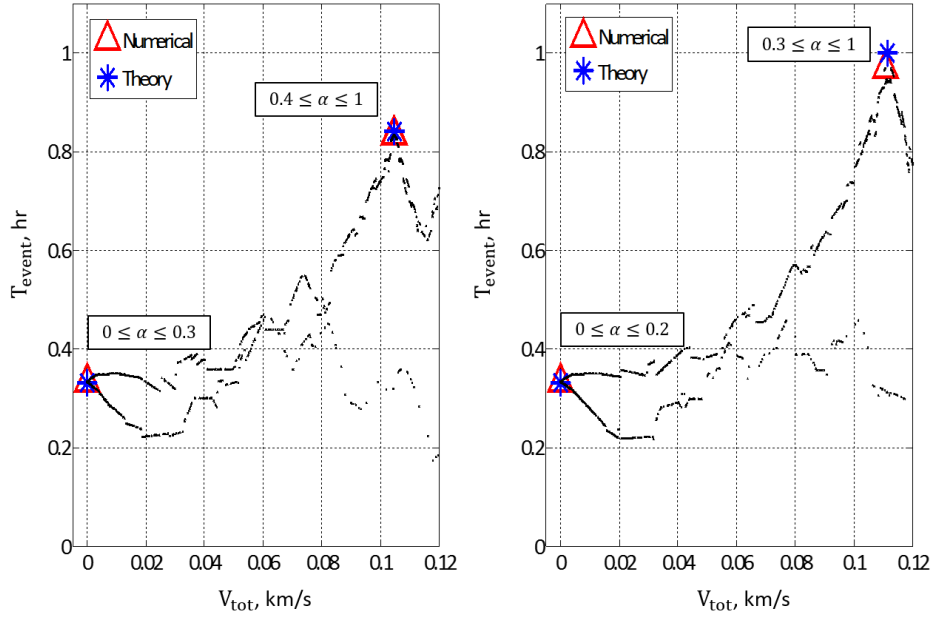


Figure 4.7: Single burn optimization with no average angle ($\beta=0$). The optimal solutions are labeled using markers and α ranges only apply for numerical solutions. Left: satellite 1. Right: satellite 2.

Note that the non-optimal solutions form the perimeter of the shape seen in Figure 4.7 and there is some agreement between the theory and numerical results. For satellite 1, the top line of the non-optimal solutions corresponds to a positive ΔV whereas the bottom line corresponds to a negative ΔV burn. However, for satellite 2, the top line corresponds to a negative ΔV burn while the bottom line is for the positive ΔV burns, see Figure 4.7. There is no smooth transition of optimal solutions for an increasing α . Instead, the optimal solutions are found in two locations, at minimum fuel cost and maximum total event time. This behavior implies that there is an alpha range that determines when either the fuel cost or the total event time is dominant in the optimization, see ranges in Figure 4.7.

Finally, based on the initial conditions, adding a single burn to the second satellite will generate a better total event time when the total event time dominates the optimization problem, see Appendix A for tables of optimal values.

Now both scenarios are presented again with the addition of the average angle inside the cost function. The cost function now has the form presented in Eq. (3.9), with β of 0.5. The same cost term parameter study is completed, see Figures 4.8-4.9.

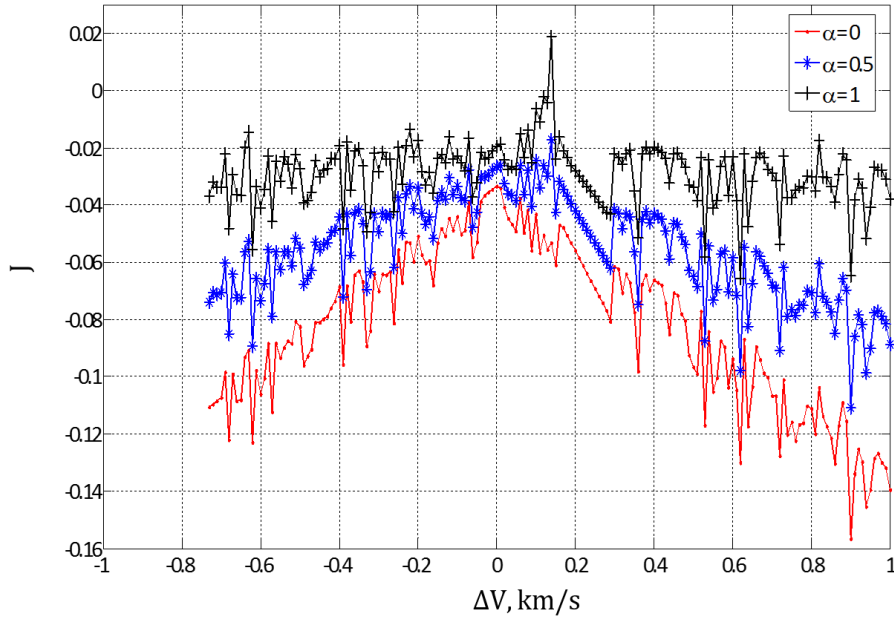


Figure 4.8: Satellite 1 single burn cost with $\beta=0.5$.

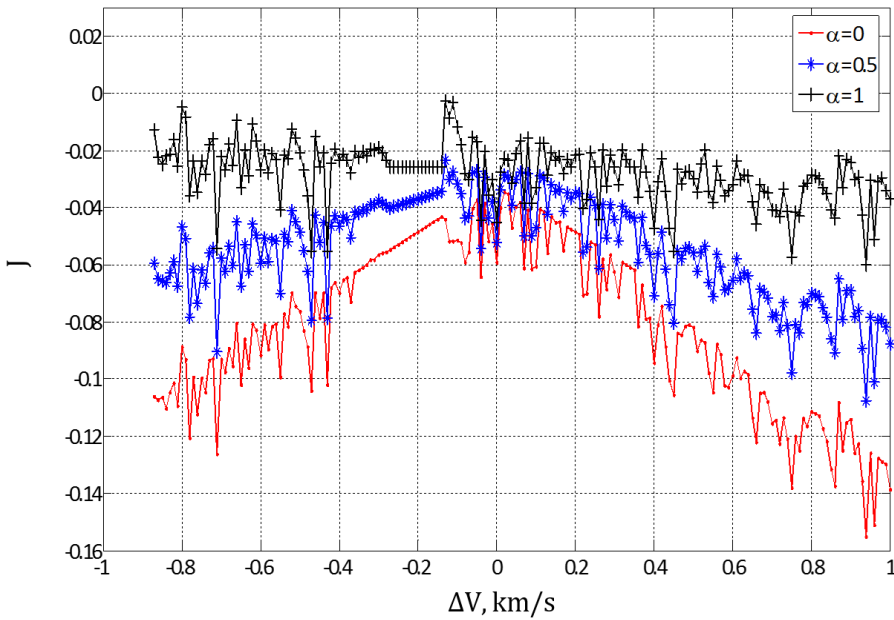


Figure 4.9: Satellite 2 single burn cost with $\beta=0.5$.

The addition of the average angle in the cost function creates additional oscillations seen in the parameter study plots. The average angle is dependent on the total event time, see Eq. (3.11), so the oscillations seen in the average angle are translated from the oscillations in the total event time, see Eq. (3.35).

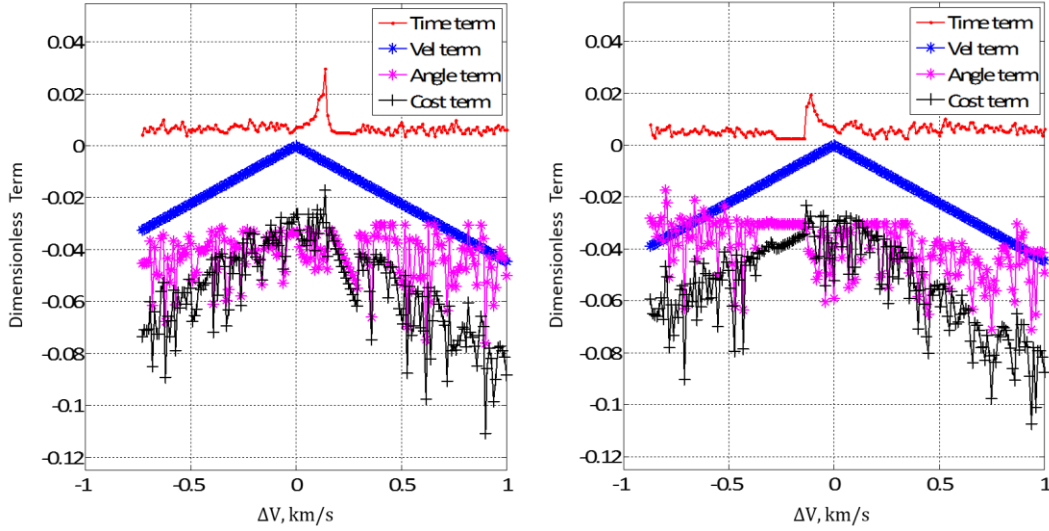


Figure 4.10: Cost function terms for $\alpha=0.5$ and $\beta=0.5$. Left: satellite 1 single burn. Right: satellite 2 single burn.

In Figure 4.10, the cost function terms are plotted for $\alpha=0.5$. The amplitude of the oscillations of the average angle is higher than that of the total event time and could create more local optimums which could decrease the effectiveness of the Global PSO algorithm. A comparison graph of the average angle calculated with the trajectory equation and with the numerical solver is shown in Figure 4.11. The magnitude of the average angle is higher in the trajectory equation calculation compared to the numerical calculation. Using the theory, ten data points are used to calculate the integral in Eq. (3.11), whereas the number of data points used in the numerical simulation may depend on the size of the time step used in the propagator. A smaller time step would allow more data points to be available, especially near the orbit apogee. However, this would significantly increase the computer running time.

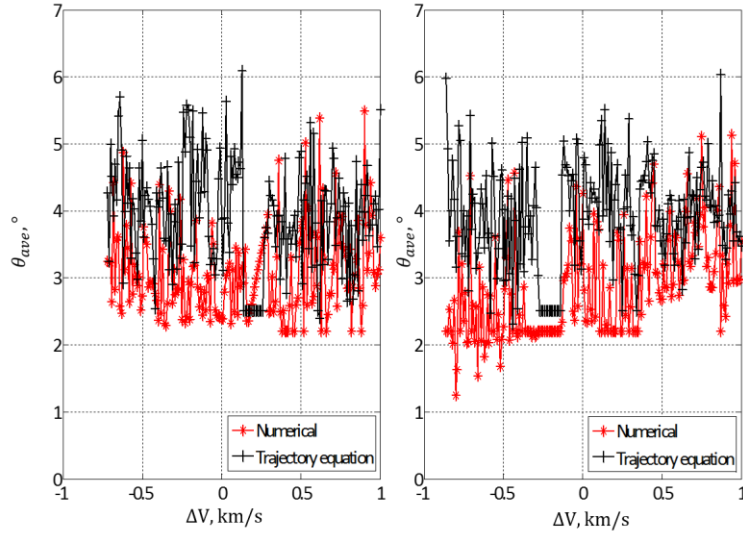


Figure 4.11: Average angle calculation for single burn case with $\alpha=0.5$ and $\beta=0.5$. Left: satellite 1 single burn. Right: satellite 2 single burn.

The optimal solutions using this form of the cost function are given in Figure 4.12. The behavior of the non-optimal solutions is similar to the previous case with no minimizing angle in the cost function. The difference between the theoretical and numerical forms is higher than previously seen. The disparity between the theoretical and numerical solutions may be found in the derivation of the average angle previously discussed. Also, the optimal solutions for an increasing α value may not correspond to the optimal time or fuel cost. Figure 4.13 gives the average angle in both scenarios with a different β value. Adding the minimizing angle term into the cost function ($\beta \neq 0$) decreases the average angle between the radius vectors of the satellites. With $\beta=0$, the average angle decreases with increasing alpha, as total event time becomes more dominant in the optimization. When $\beta=0.5$, the average angle for both satellites increases with an increasing alpha value.

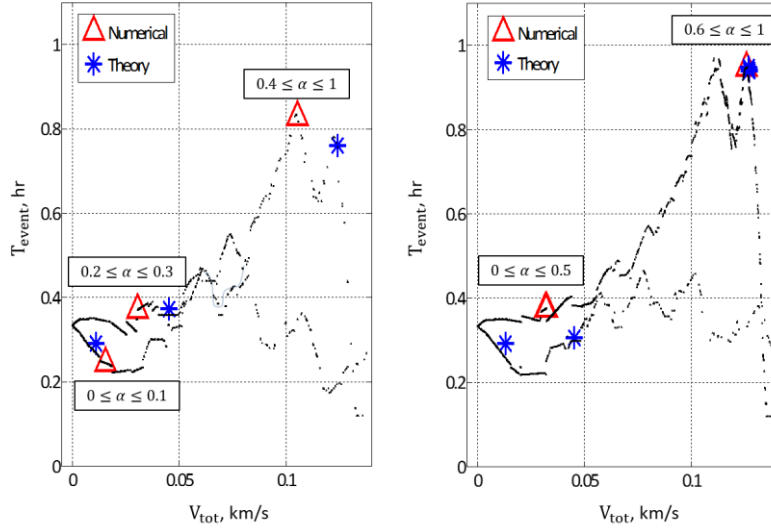


Figure 4.12: Single burn optimization with average angle included ($\beta=0.5$). The optimal solutions are labeled using markers and α ranges only apply for numerical solutions. Left: satellite 1. Right: satellite 2.

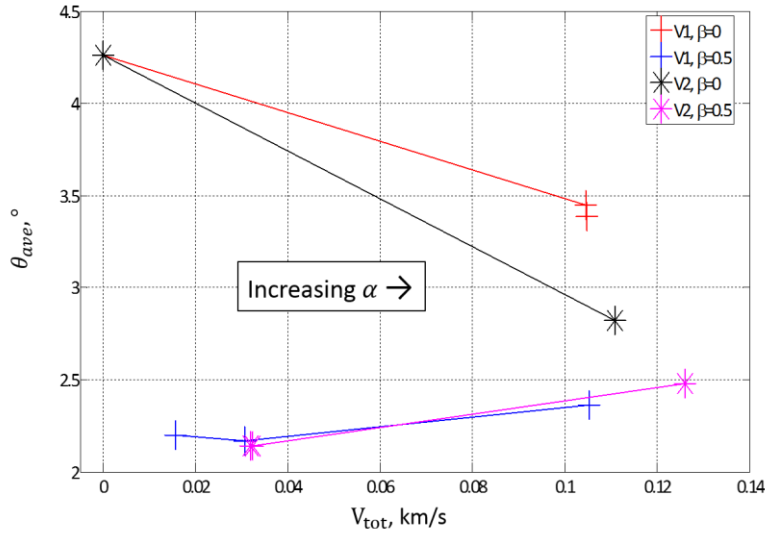


Figure 4.13: Average angle comparison of optimal solutions for single burn optimization with unperturbed motion.

4.1.2 Double Burn Optimization

A single independent instantaneous burn is applied to each satellite at their respective perigee location. The optimal configuration is found by using Eq. (4.1) with the same variable α values and constants. The parameter study for some of these cases is shown in Figure 4.14. Similar to the single burn case, the case with $\alpha=0$ would give an optimal fuel cost solution while using $\alpha=1$

would give an optimal total event time. For an increasing alpha, the optimal solution is defined in the region of high negative burns (deceleration burns). Figure 4.15 presents the velocity and the time dimensionless terms in the cost function. The locations of the optima are different and, as seen in Figure 4.14, these differences cause the global optimum to move for an increasing α .

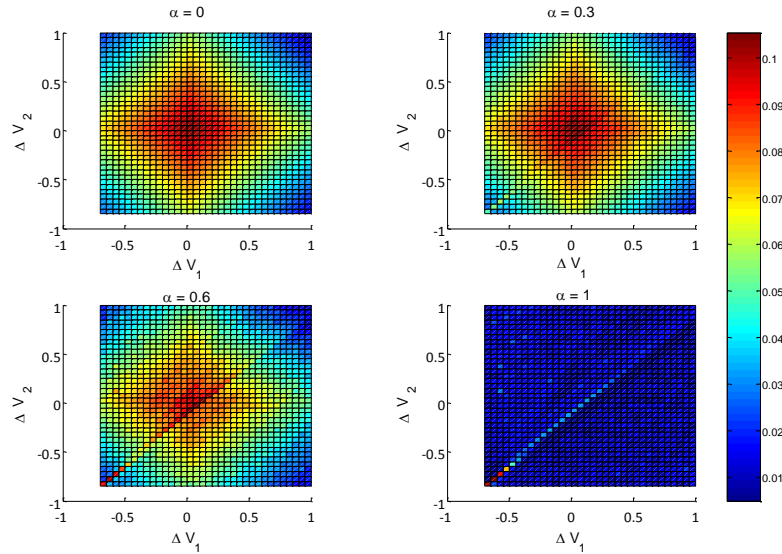


Figure 4.14: Surface of cost function with no average angle ($\beta=0$).

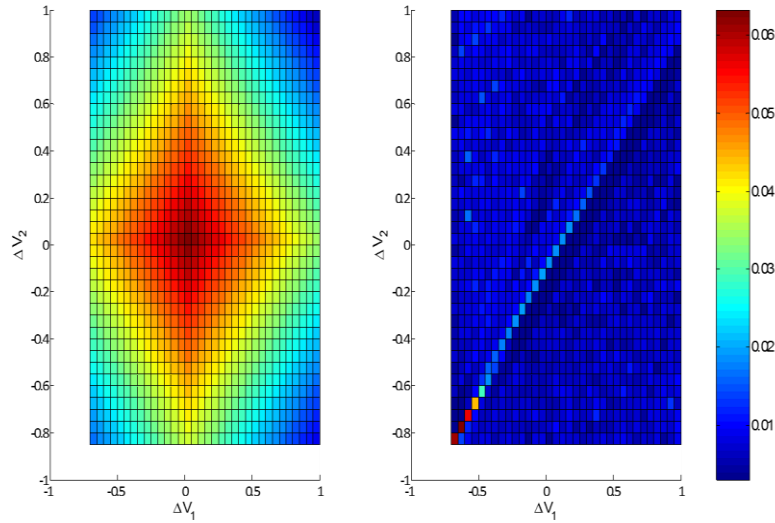


Figure 4.15: Left: velocity term. Right: total event time term.

A pareto front of the optimal solutions is given in Figure 4.16. Similar to the single burn optimization, the transition of the optimal solution is not smooth and appears in the form of

jumps. In this case, there are two jumps which signify the transition from fuel cost dominant to event time dominant. The non-optimal solutions are contained within the shaded region of Figure 4.16. The optimal solutions for low and high α values corresponds to the minimum fuel cost and maximum total event time locations in the pareto front, respectively. However, now there is a region of α values ($0.3 \leq \alpha \leq 0.6$) in which the optimal solutions are located at neither the optimal fuel cost nor total event time. This alpha range corresponds to the region in which the impact of the fuel cost and the time on the optimization are equal, see Appendix A for the list of optimal solutions.

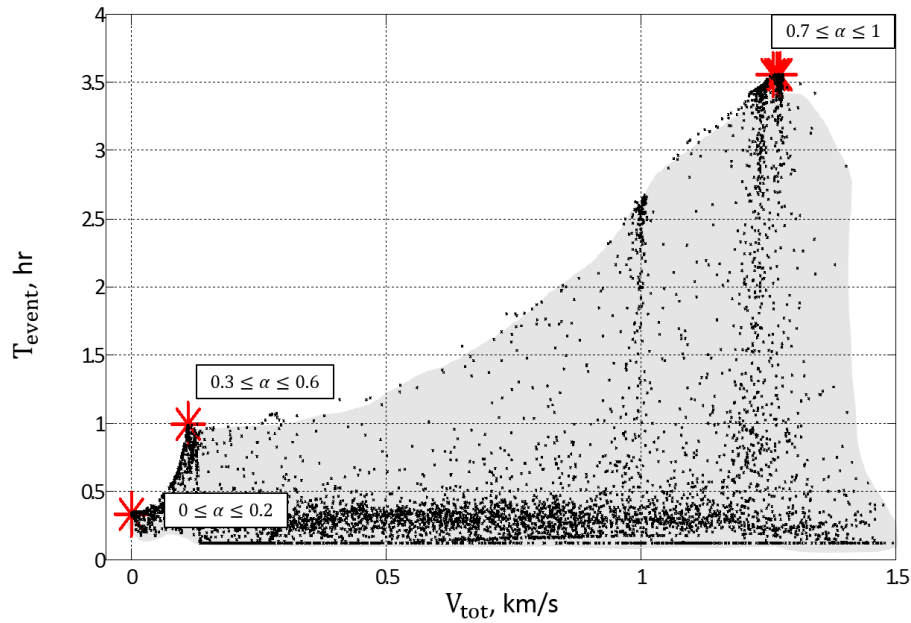


Figure 4.16: Pareto front for double burn optimization with no average angle ($\beta=0$). Optimal solutions are labeled in red with their appropriate α ranges.

Next, a cost function, Eq. (3.9), that includes the average angle ($\beta=0.5$) is used in the double burn optimization problem. The same parameter study is shown in Figure 4.17 and the average angle term and the cost function for $\alpha=0.6$ is given in Figure 4.18. The behavior of the cost function given a variable α is similar as before, but there are now more prominent oscillations in the cost surface. These oscillations are a function of the average angle, see Figure 4.18.

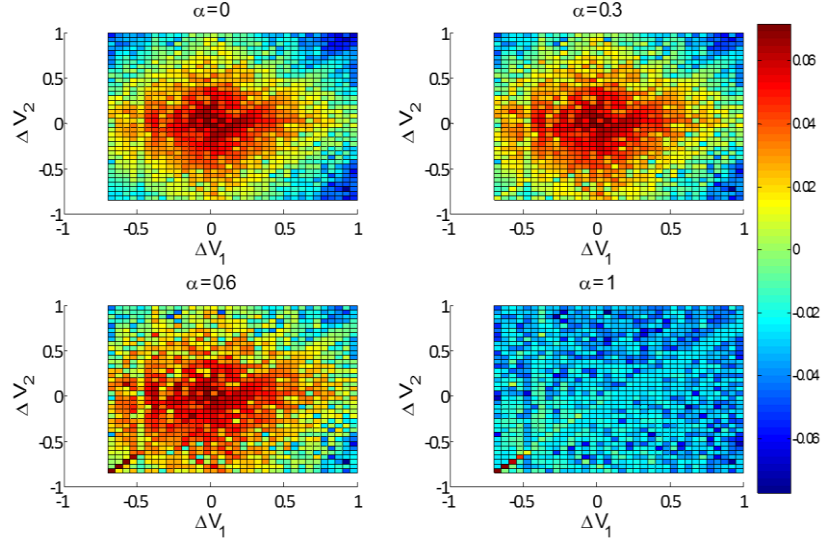


Figure 4.17: Cost value for double burn optimization with average angle ($\beta=0.5$).

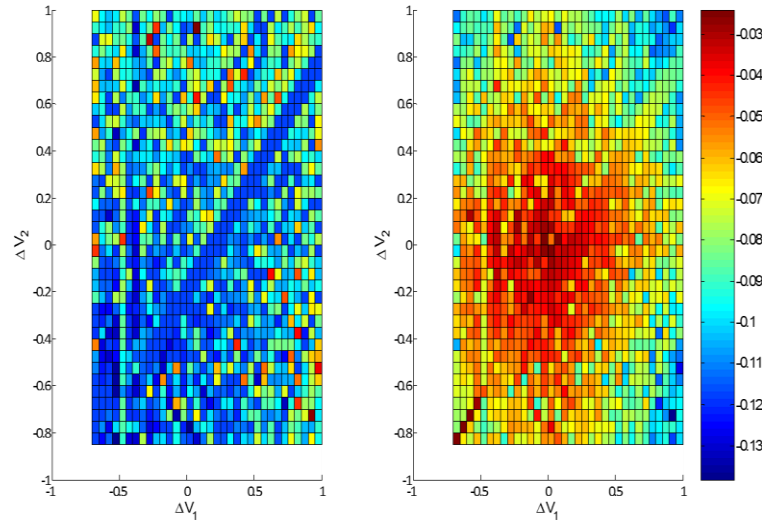


Figure 4.18: Function terms for $\alpha=0.6$. Left: Dimensionless average angle term. Right: Dimensionless cost term.

Finally, a pareto front is presented with the addition of the average angle ($\beta=0.5$), see Figure 4.19. Contrary to the previous double burn case, the optimal solutions are now arranged at four different locations. Three of these locations are similar to the previous case, but there is now an optimum that is located near maximum total event time. Looking at Figure 4.20, it can be inferred that this optimum is the minimum average angle result, at $\Delta V_{\text{tot}} = 1.09$ km/s. Notice that this value is located near the optimal total event time which is a result of how the average angle

is defined, see Eq. (3.11). A comparison of the average angles is given in Figure 4.20. The average angle is smaller with the addition of a nonzero β value in the cost function. Similar to single burn case, the average angle has a negative slope as α increases. This is expected because the definition of the average angle contains the total event time definition in the denominator and as α increases so does the total event time in the optimal solution. However, at high α , the optimum solution is located near the maximum total event time solution for both cases. This is because the optimization is giving priority to the total event time rather than the average angle and would explain the increase of the average angle when $\alpha=1$ and $\beta=0.5$, see Appendix A for table of optimal solutions.

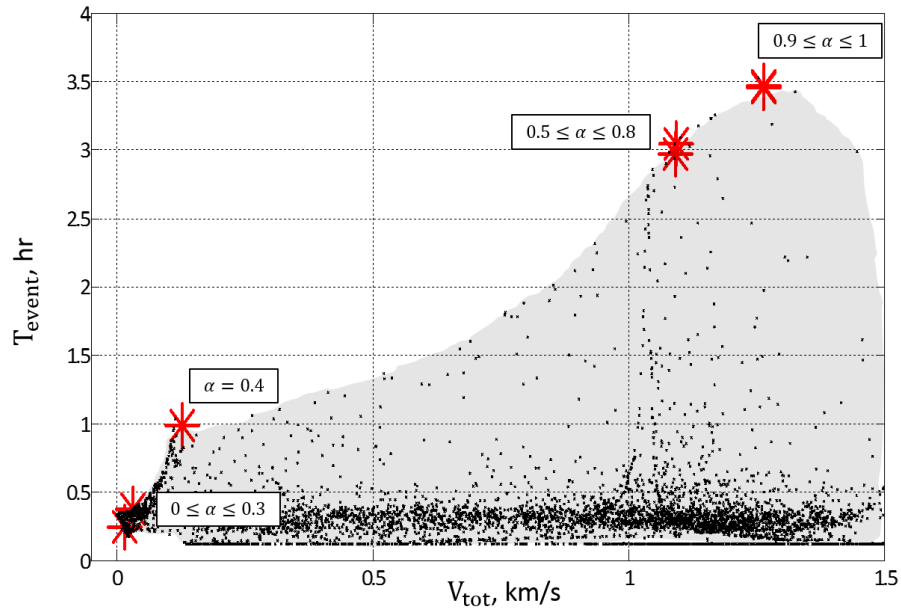


Figure 4.19: Pareto front of double burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled in red with corresponding α ranges.

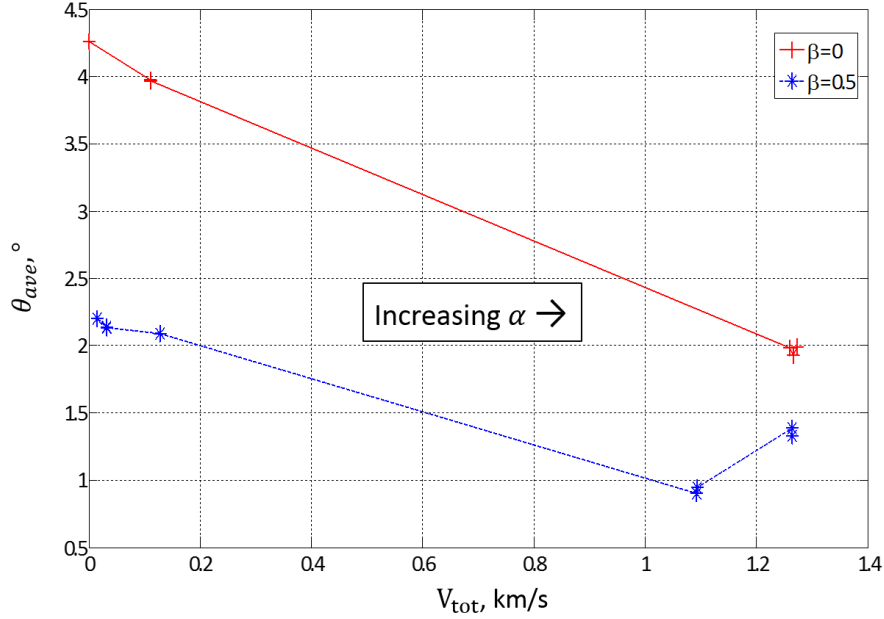


Figure 4.20: Average angle of optimal solutions with ($\beta=0.5$) and without ($\beta=0$) its inclusion in the double burn optimization.

4.1.3 Four Burn Optimization

Two instantaneous burns are applied to both satellites during the mission. The burns are independent and are applied at different times along the propagation, see Figure 3.5. The pareto front of the optimal solutions using Eq. (4.1) with no average angle is given in Figure 4.21. The optimal solutions form the same pattern as in the previous burn scenario. There are three different locations where the optimal solutions are located: at minimum fuel cost, at an intermediate location, and near maximum total event time. Similar to before, the intermediate solution is given when the impact of the time and the fuel cost terms in the cost function are equal in the optimization. The non-optimal solutions are displayed in the shaded area. Note that that optimal solutions are located in the perimeter of the possible shaded solutions. Also, with an increased problem dimension, the PSO algorithm version used has a difficult time converging to the true optimum for high α values which is why the optimal solutions are more spread out.

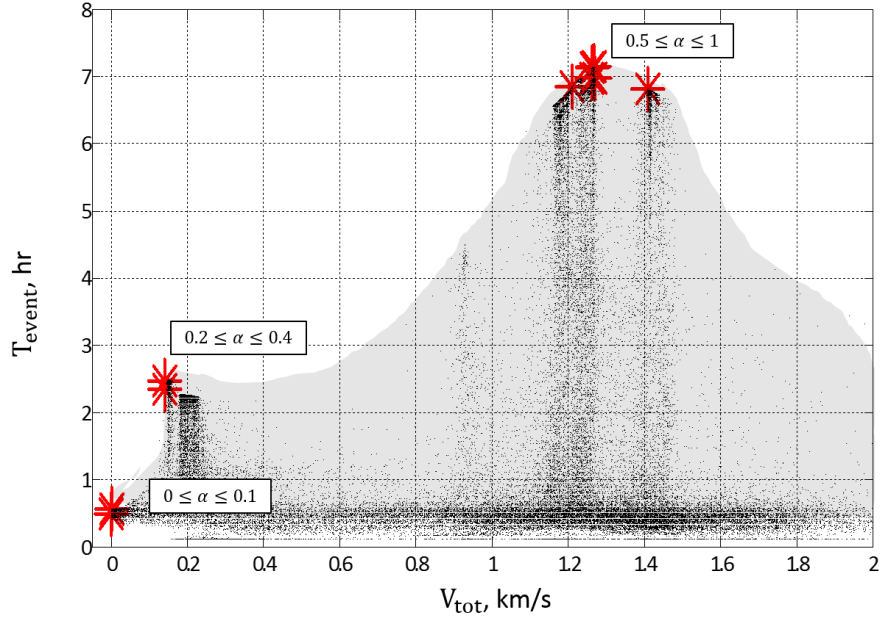


Figure 4.21: Pareto front of four burn optimization case with no average angle included ($\beta=0$). Optimal solutions are labeled in red with corresponding α ranges.

Using Eq. (3.9) and including the average angle in the optimization ($\beta=0.5$), the pareto front is displayed in Figure 4.22. The location of the optimal solutions differ from the previous case. At high α , the optimal solution may not be located at a maximum total event time, but at a region of otherwise non-optimal solutions in the previous case due to the inclusion of the average angle. Figure 4.23 implies that the average angle decreases with the addition of a nonzero β value in the cost function. The increase in α also decreases the average angle in the optimal solution.

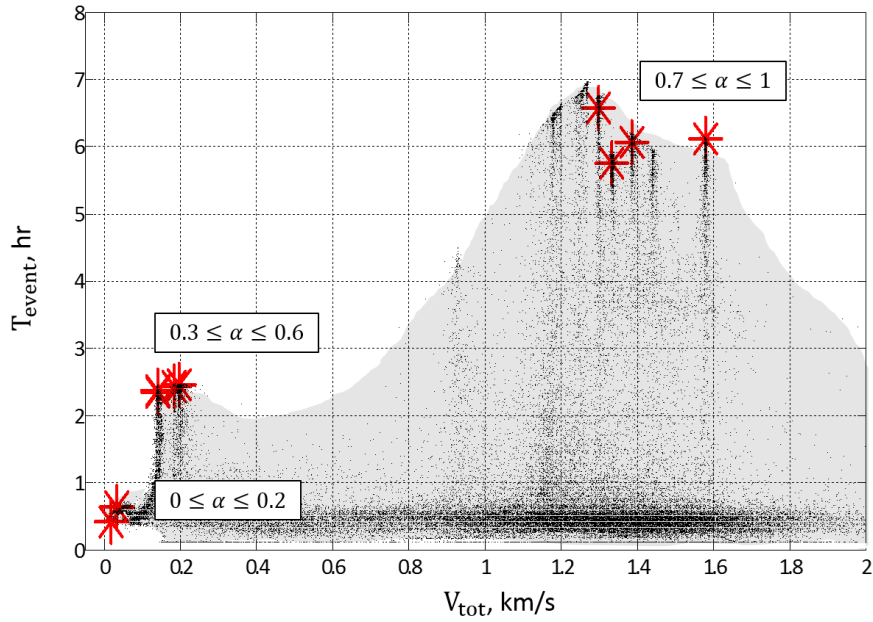


Figure 4.22: Pareto front of four burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled in red with corresponding α ranges.

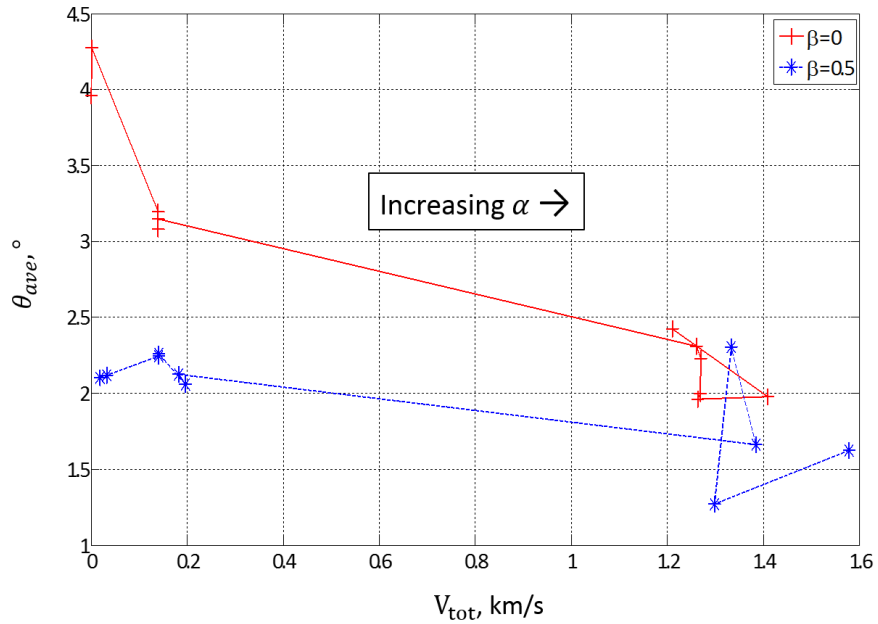


Figure 4.23: Average angle of optimal solutions with ($\beta=0.5$) and without ($\beta=0$) its inclusion in the four burn optimization.

4.1.4 Summary

In this section, optimal solutions are given for an optimization of three different cases with an unperturbed two-body orbital mechanics model. For a single burn case, an applied burn to the

second satellite would give a higher total event time for the mission for the same fuel cost. Optimal solutions in the single burn case satisfy either the time or fuel cost objective, but there are no specific optimal solutions that optimizes both. Applying a single burn to each satellite increases the total event time given by the optimization. In this case, there is an optimal solution that satisfies both the time and fuel cost objectives, for $\Delta V_{\text{tot}} \approx 0.1113$ km/s. For the case with four burns, a layout of optimal solutions similar to the double burn case is given. Although, now with an increased problem dimension, the difference in the optimal solutions is greater. The PSO has a difficult time trying to locate optimal solutions for high α values, probably due to increased oscillations in T_{event} . The addition of the average angle term in the cost function for all cases does decrease the average angle between the satellites radius vectors. However, this addition may decrease the T_{event} possible for the optimal solutions.

4.2 Two-Body Motion with J_2 Effect

In this section, the same three instantaneous burn cases are analyzed using a perturbed orbital dynamics model, see Section 2.3. Pareto fronts of the optimal solutions are presented along with a comparison to the unperturbed case. Optimal results with β values of 0 and 0.5 in Eq. (3.9) are compared and used to justify the inclusion of the average angle term.

4.2.1 Single Burn Optimization

A single burn is applied to a single satellite using a perturbed orbital dynamics model. Two scenarios are included: a single burn is applied only to satellite 1 and a single burn is applied to satellite 2. The cost function is in the form of Eq. (4.1) with no average angle included and with a variable α value of $\alpha = \{0, 0.1, \dots, 1\}$. First, a parameter study to explore the physics of the problem using perturbed orbital dynamics model is completed, see Figure 4.24.

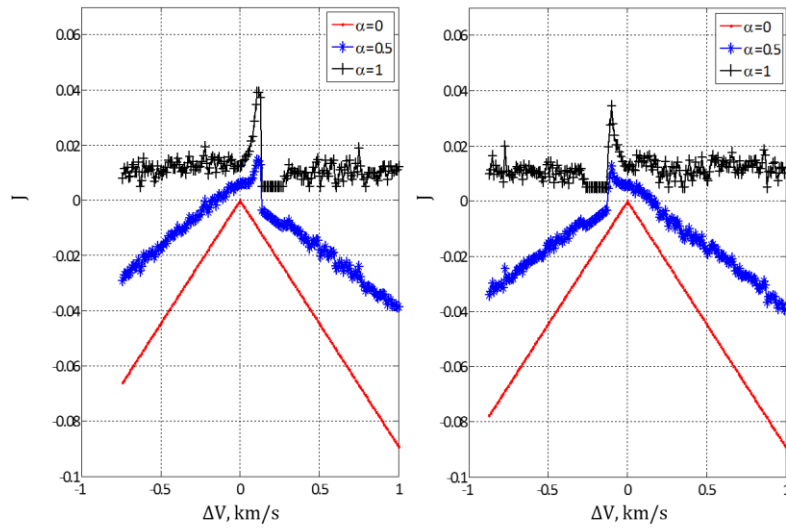


Figure 4.24: Cost of J_2 effect single burn optimization. Left: satellite 1 burn only. Right: satellite 2 burn only.

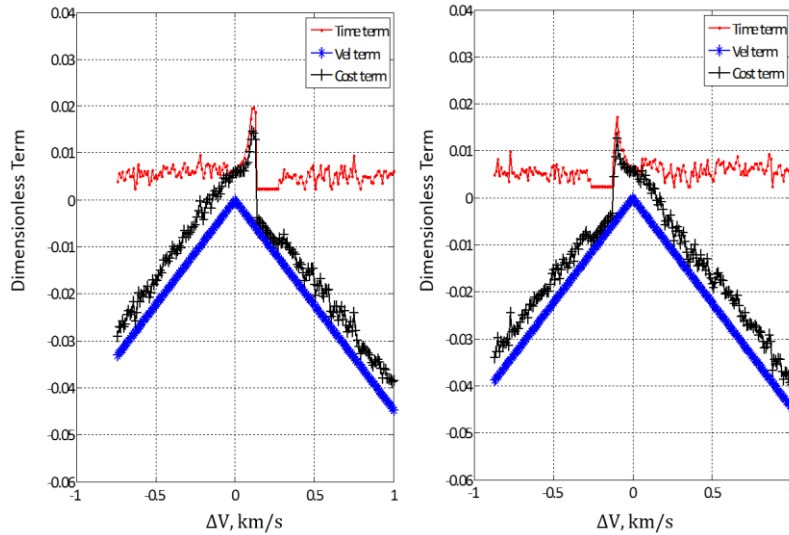


Figure 4.25: Cost terms single burn optimization with $\alpha=0.5$. Left: satellite 1 burn only. Right: satellite 2 burn only.

Figures 4.24-4.25 show that the behavior of the cost function and the cost terms for both scenarios is similar to the unperturbed case. The oscillations that results in the cost term are caused by the oscillating behavior of the total event time. This behavior may also be explained in the same manner with the time and number of events presented in Figure 4.6. Figure 4.26 shows the optimal solutions for both scenarios. All solutions occur in the perimeter of the shape formed in Figure 4.26. Similar to the unperturbed case, the optimal solutions are located at the locations

of minimum fuel cost and maximum total event time. This also implies that there are two regions in which the fuel cost and event times are dominant in the optimization problem. Also, the addition of the J_2 perturbation decreases the maximum total event time possible. In Figure 4.26, satellite 1 has positive ΔV burn solutions in the top line and negative ΔV burn solutions in the bottom line. In contrast, satellite 2 has negative ΔV burn solutions in the top line and positive ΔV burn solutions in the bottom line.

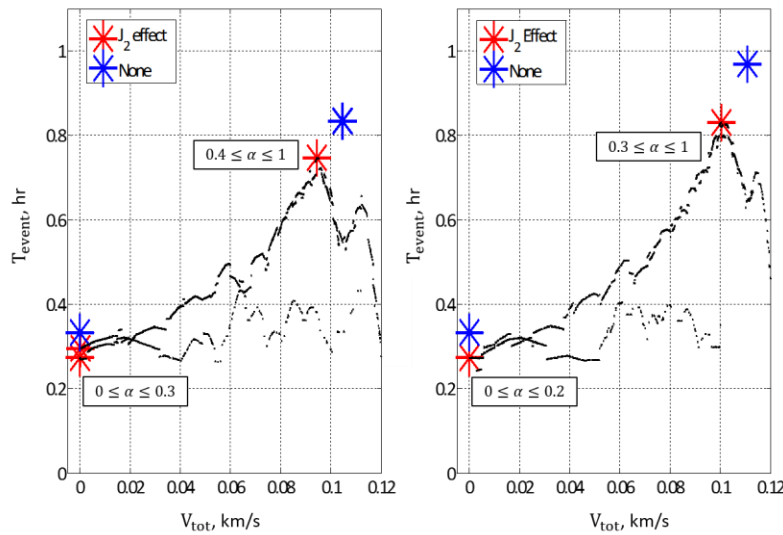


Figure 4.26: J_2 effect single burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply to J_2 effect solutions. Left: satellite 1 burn only. Right: satellite 2 burn only

Now, using Eq. (3.9) with the average angle ($\beta=0.5$), the optimal solutions can be presented in Figure 4.27. For the satellite 1 single burn, there are three distinct optimal location. Each corresponds to an area of α when: the fuel cost is optimal, the average angle is optimal, and the total event time is optimal. This can also be observed in Figure 4.28, where the second optimal solution corresponds to the lowest average angle value. The behavior of the optimal solutions is similar to that of the unperturbed results. In the satellite 2 single burn, there are two optimal locations, similar to the unperturbed results. These solutions also correspond to areas where the fuel cost and time would be dominant over the optimization. Also, from Figure 4.28, the location

of optimal total event time might also be the location of optimal average angle. Note that the location of these optimal solutions are not located at minimum fuel cost or maximum total event time because of the inclusion of the average angle term, see Figure 4.27.

Figure 4.28 shows that the average angle does decrease for a nonzero value of β in the cost function and that the optimal average angle is found with significantly different fuel costs. Also, for the cost function with no average angle included and defined by Eq. (4.1), the average angle in the optimal solution increases as the optimization variable α increases.

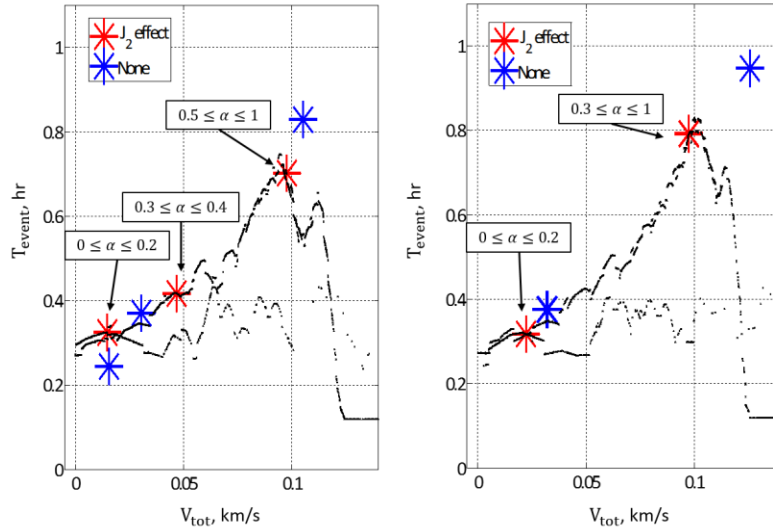


Figure 4.27: J_2 effect single burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply to J_2 effect solutions. Left: satellite 1. Right: satellite 2.

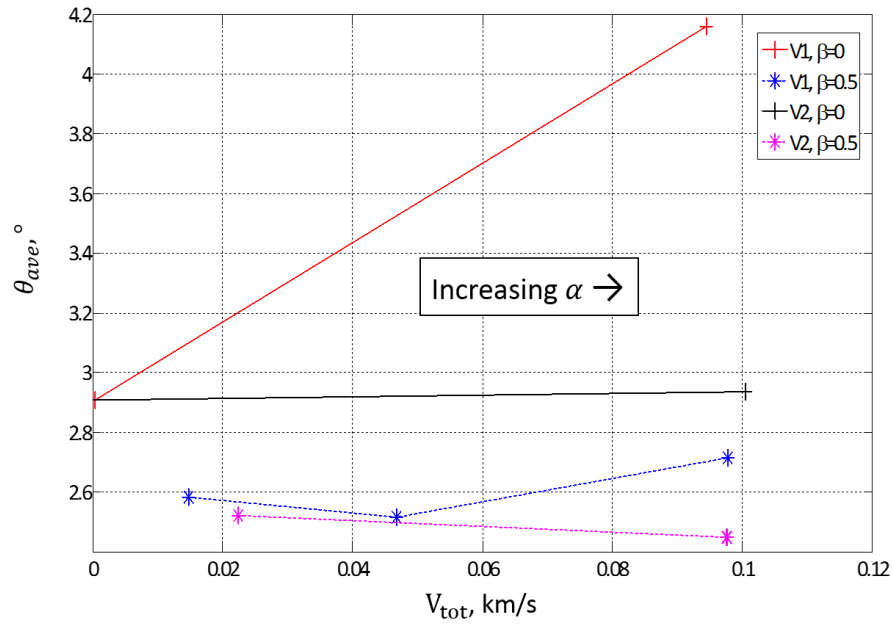


Figure 4.28: Average angle of optimal solutions comparison of single burn optimization with J_2 effect. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.

4.2.2 Double Burn Optimization

A single burn is applied to each satellite at their respective perigee using a perturbed orbital mechanics model. The same parameter study can be performed with Eq. (4.1) and a variable α , see Figure 4.29, and Eq. (3.9) with $\beta=0.5$, see Figure 4.30. The surfaces for the perturbed case, with a β value of 0 and 0.5, have the same behavior as the unperturbed case. Slowly incrementing α changes the dominant term in the cost function.

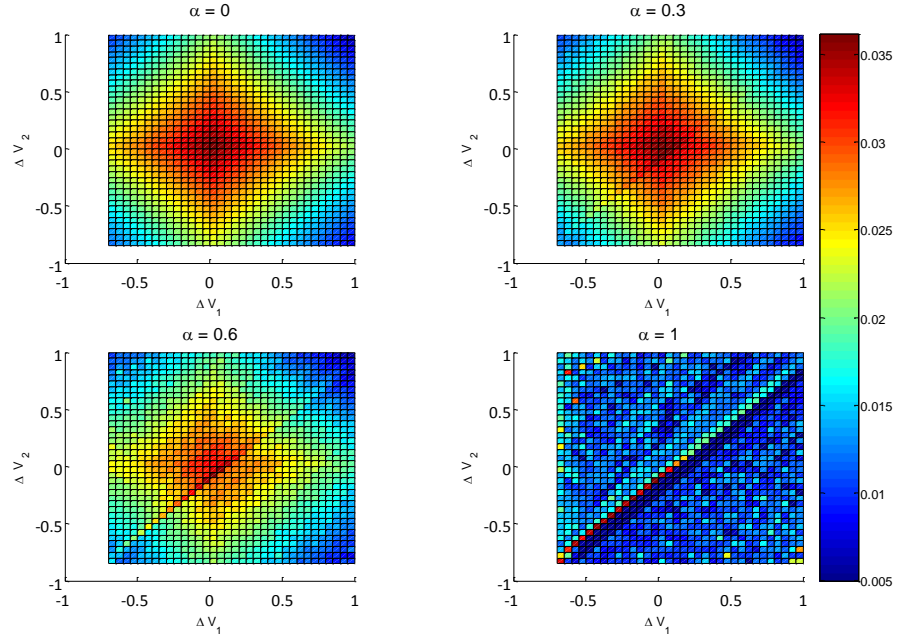


Figure 4.29: Cost surface with variable alpha for $\beta=0$.

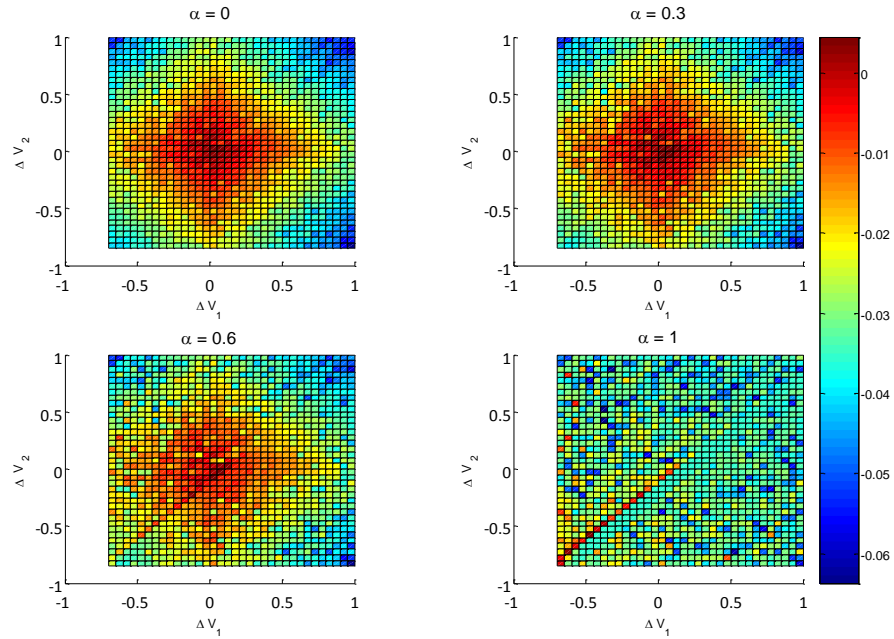


Figure 4.30: Cost surface with variable alpha for $\beta=0.5$.

The pareto front for both β values are presented in Figures 4.31-4.32 with the shaded area representing the non-optimal solutions generated by the PSO algorithm. For

$\beta=0$, the locations of the optimal solutions are in three different locations: at minimum fuel cost, near maximum total event time, and at $V_{\text{tot}} \approx 0.1$ km/s. This behavior is similar to the unperturbed case. With $\beta=0.5$, the optimal solutions are found near the same locations as the case with a $\beta=0$. Figure 4.33 shows the decrease in average angle with a nonzero β in the cost function. Similar to the unperturbed case, the average angle decreases for an incrementing α value in the cost function. At high α values, the optimal solution for both cases are located at maximum total event time.

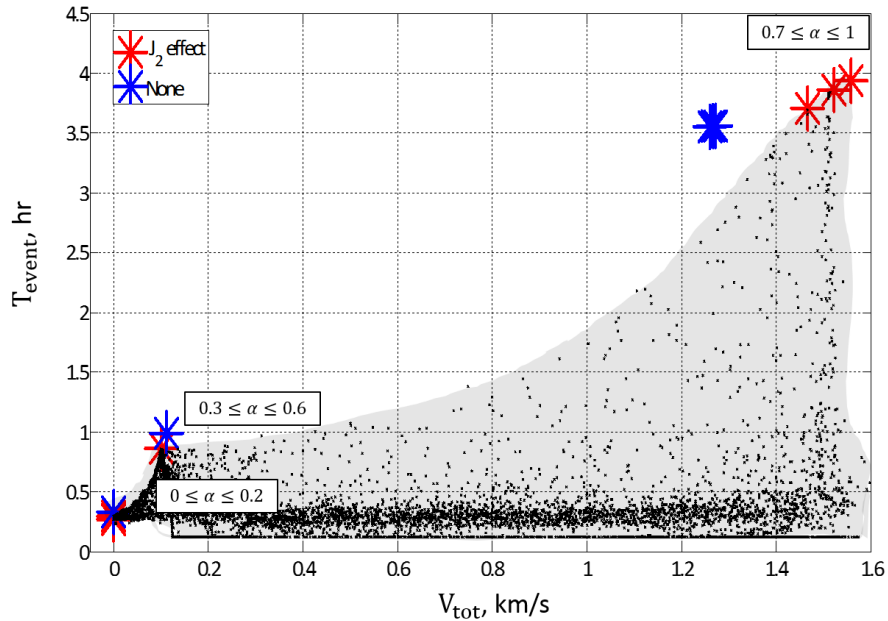


Figure 4.31: Pareto front of J_2 effect double burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply for J_2 effect optimization.

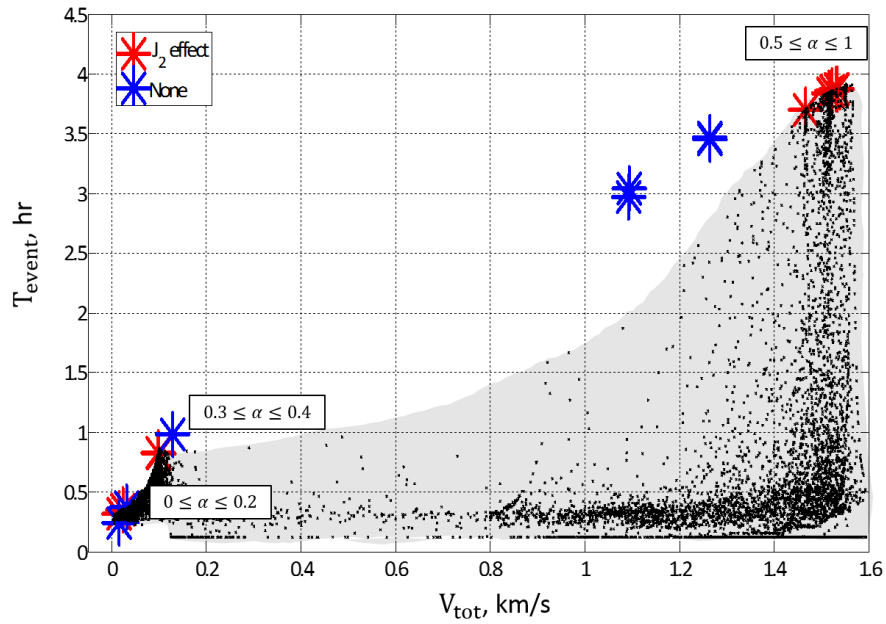


Figure 4.32: Pareto front of J_2 effect double burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply for J_2 effect optimization.

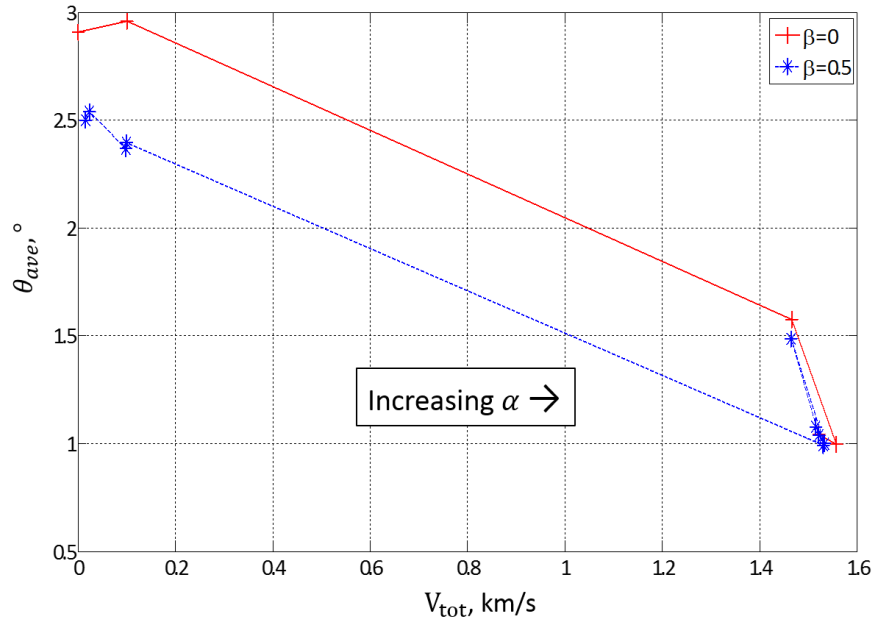


Figure 4.33: Angle comparison of optimal solutions of double burn optimization with J_2 effect. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.

4.2.3 Four Burn optimization

Two burns are applied to each satellite at their perigee locations using a perturbed orbital mechanics model. Two cases are presented with a cost function shown in Eq. (3.9) with a β value

of 0 and 0.5. The pareto front of the first case is presented in Figure 4.34. Similar to unperturbed case, the non-optimal solutions are formed inside the shaded area and the optimal solutions are located at three different locations: at minimum fuel cost, at maximum total event time, and at $V_{\text{tot}} \approx 0.137 \text{ km/s}$ ($V_{\text{tot}} \approx 0.144 \text{ km/s}$ for $\beta=0.5$). Figure 4.36 implies that the average angle decreases with a nonzero β in the cost function. The same behavior is seen in the unperturbed case.

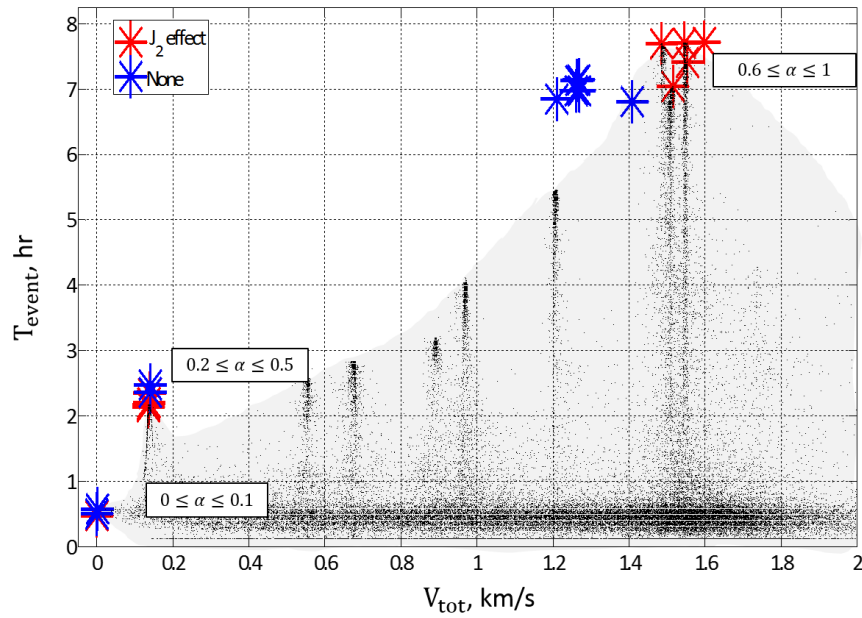


Figure 4.34: Pareto front of J_2 effect four burn optimization with no average angle included ($\beta=0$). Optimal solutions are labeled with markers and α ranges only apply for J_2 optimization.

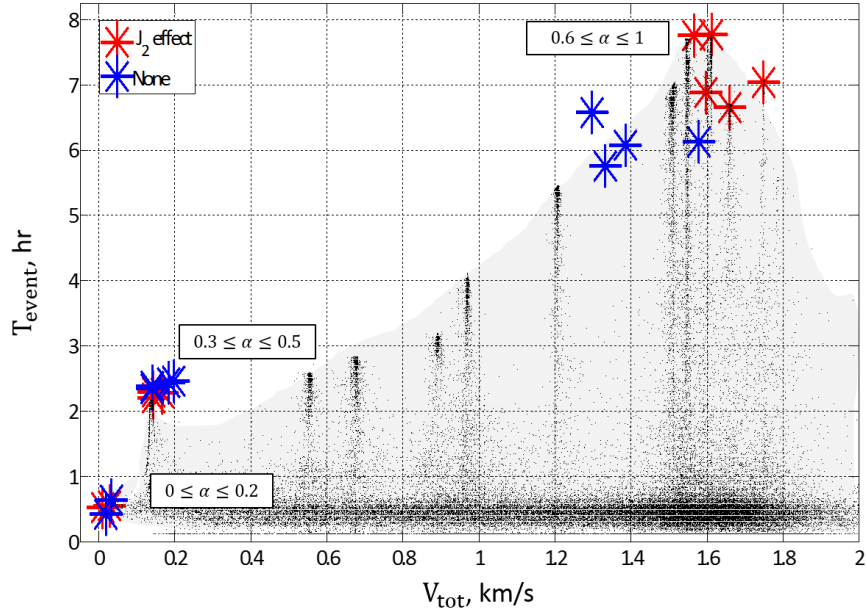


Figure 4.35: Pareto front of J_2 effect four burn optimization with average angle included ($\beta=0.5$). Optimal solutions are labeled with markers and α ranges only apply for J_2 optimization.

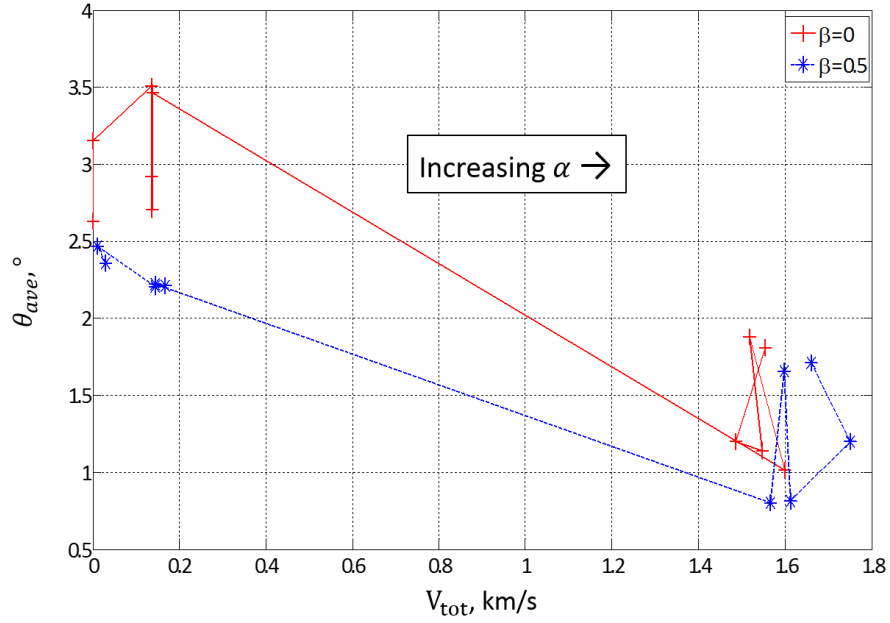


Figure 4.36: Angle comparison of optimal solutions of J_2 effect four burn optimization. Optimal solutions are obtained with ($\beta=0.5$) and without ($\beta=0$) the average angle in the cost function.

4.2.4 Summary

In this section, the optimal solutions are obtained using a perturbed two-body problem in the propagator inside the cost function. The results are compared to the results given in the

unperturbed two-body problem case. In the perturbed case, which is more accurate to real simulations, the possible total event time decreases. These changes may be caused by the secular variations to the orbital parameters, ω and Ω , see Figures 4.37-4.38. Although all the same behavior patterns are seen as in the unperturbed case. The addition of the average angle term in the cost function does decrease the average angle in the optimal solution. Again, the addition of this angle also decreases the T_{event} possible and increases the fuel cost needed.

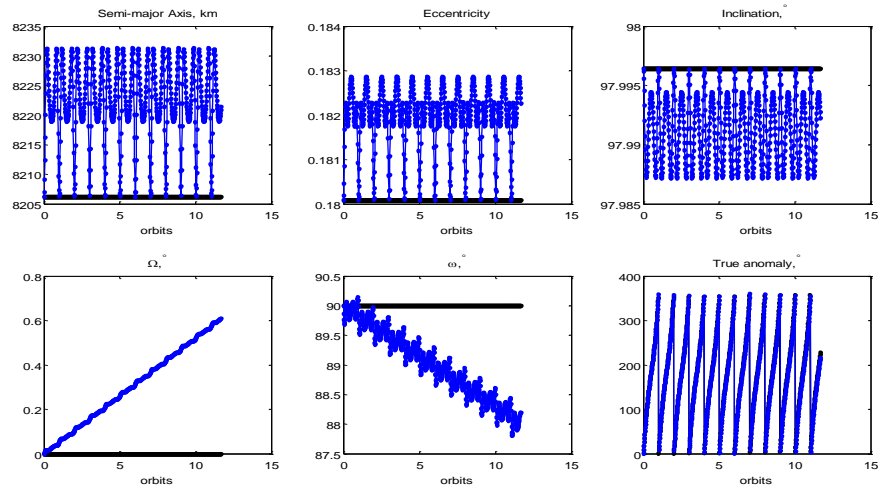


Figure 4.37: J_2 effect on satellite 1 under initial conditions.

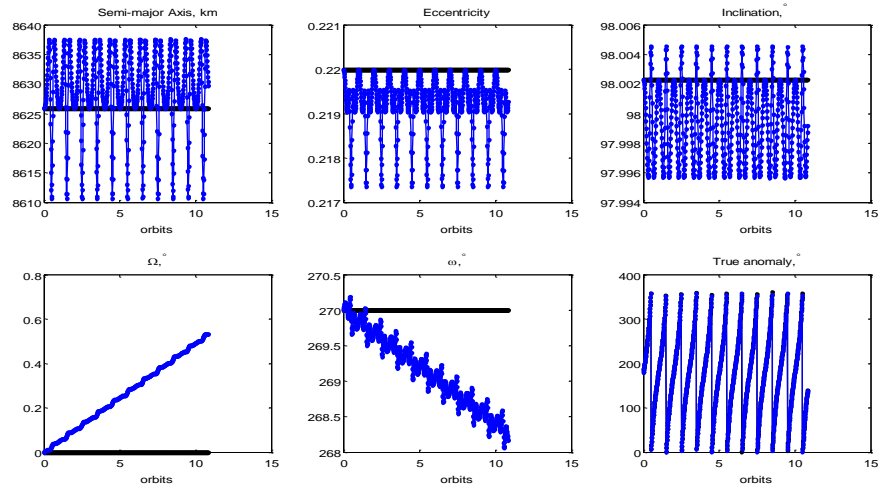


Figure 4.38: J_2 effect on satellite 2 under initial conditions.

Chapter 5

Conclusions

The purpose of this thesis was to explore a two satellite orbit design that could be used to collect data inside a certain geomagnetic latitude range over the Earth. Two satellites would be used to collect data on the particle interactions that occur inside the auroral oval which may cause the formation of the auroral displays. A two satellite system is beneficial for simultaneously studying the particle acceleration that is responsible for the formation of the aurora at different altitudes over the same geomagnetic latitude region. In this thesis, a series of burns were applied to generate an orbit for both satellites that could maximize the time that both satellites see each other over the auroral ovals. This was achieved by an analysis of the optimum solutions generated by a Particle Swarm Optimization method using a cost function with variable weights on the total event time and fuel cost terms. A variable α was defined in the cost function as a weight constant in the range of $0 \leq \alpha \leq 1$. In addition, a term correlated to the average angle between the radius vectors of both satellites is introduced in the cost function. A number of conclusions can be made using these cost function definitions in different burn scenarios.

The scenarios analyzed in this thesis include the following: a single burn case, a double burn case, and a four burn case. The orbital mechanics model used was a simple unperturbed two-body problem. Incrementing α in the cost function generated optimal solutions which were

associated with the optimal total event time and fuel cost in all three scenarios. Although in the latter two scenarios, there was an alpha range in which the optimal solution wasn't associated with either optimal total event time or fuel cost. The addition of the average angle in the cost function generated optimal solutions that were not associated with any previous optimal time or fuel cost. Additionally, in the double burn and four burn case, a new optimal solution was generated that was associated with the minimum average angle.

The Earth's equatorial bulge perturbation, J_2 effect, is added to the orbital mechanics model. The same three scenarios are analyzed with the same cost function in which the behavior of the optimal solutions is observed to be similar to the unperturbed case. However, the total event time of the optimal solutions is decreased due to the secular effects caused by the J_2 perturbation. The possible additions of other perturbations, such as the atmospheric drag or solar radiation pressure, could further reduce the total time of any optimal solutions.

The cost function given in this paper with weight constants of $\alpha=0.3$ and $\beta=0.5$ gives the "best" optimal results in all burn cases explored. None of the terms in the cost function are particularly dominant over the others and all objectives are satisfactorily fulfilled. The average angle term is an important addition to the cost function because it increases the potential data periods in which the satellites line up over the same position over the Earth's surface, same latitude and longitude but separate altitudes.

5.1 Future Work

In the previous section it was recommended to use the following weight constants in the cost function, $\alpha=0.3$ and $\beta=0.5$. Additional studies with an odd number of burns can be explored with this new cost function, e.i. two burns are applied to satellite 1 and one burn is applied to satellite 2. The results generated are only applicable to the initial conditions chosen by the author. For

future studies, it is recommended that weight constants α and β be chosen beforehand and that the initial orbits be varied to study the best initial orbit to place these satellites. The addition of the J_2 perturbation has been shown to decrease the total event time available from the orbital configuration. It would be beneficial to increase the number of possible perturbations in order to better simulate the space environment the satellites will experience in LEO. The additional perturbations may include but are not limited to: higher order zonal harmonics, atmospheric drag, solar radiation, third body effects from the moon, solar tide effects. Finally, varying both α and β in the cost function to generate a surface of optimal solutions, similar to the pareto front given, would help choose possibly better weight constants for the fuel cost, total event time, and average angle.

Future studies can be completed by using a different dual satellite mission. This new mission would focus on the electron accelerations along the same magnetic flux in the Earth's magnetic field. Simultaneous data would be recorded by both satellites along the same magnetic flux, but at different altitudes. The new objective of this mission would be to optimize the instances in which both satellites are in the same magnetic flux.

Bibliography

1. Kelley, M.C., *The Earth's Ionosphere: Plasma Physics & Electrodynamics*. 2009: Elsevier Science.
2. Sellers, J.J., et al., *Understanding Space*. 2003: McGraw-Hill Custom Pub.
3. *Earth's Magnetosphere*. 2014; Available from: <http://www-ssc.igpp.ucla.edu/personnel/russell/papers/magsphere/msphere06.gif>.
4. Lee, S., et al., *Geomagnetic conjugate observations of plasma-sheet electrons by the FAST and THEMIS satellites*. Journal of Geophysical Research: Space Physics, 2013. **118**(1): p. 132-145.
5. Colpitts, C.A., et al., *Simultaneous ground and satellite observations of discrete auroral arcs, substorm aurora, and Alfvénic aurora with FAST and THEMIS GBO*. Journal of Geophysical Research: Space Physics, 2013. **118**(11): p. 2013JA018796.
6. Carlson, C.W., R.F. Pfaff, and J.G. Watzin, *The Fast Auroral SnapshoT (FAST) Mission*. Geophysical Research Letters, 1998. **25**(12): p. 2013-2016.
7. Kovalic, T. *NASA's Polar Mission: Unlocking the Secrets of Earth's Magnetosphere*. 2006 [cited 2014 August 1]; Available from: <http://pwg.gsfc.nasa.gov/polar/>.
8. Netting, R. *Polar*. 2014 May 13 [cited 2014 August 1]; Available from: <http://science.nasa.gov/missions/polar/>.
9. Janhunen, P., et al., *Alfvénic Electron Acceleration in Aurora Occurs in Global Alfvén Resonosphere Region*. Space Science Reviews, 2006. **122**(1-4): p. 89-95.
10. Hultqvist, B., *The swedish satellite project Viking*. Journal of Geophysical Research: Space Physics (1978–2012), 1990. **95**(A5): p. 5749-5752.
11. Lindqvist, P.A. and G. Marklund, *A statistical study of high-altitude electric fields measured on the Viking satellite*. Journal of Geophysical Research: Space Physics (1978–2012), 1990. **95**(A5): p. 5867-5876.
12. Lundin, R., G. Haerendel, and S. Grahm, *The FREJA project*. Geophysical research letters, 1994. **21**(17): p. 1823-1826.
13. Karlsson, T., et al., *Subauroral electric fields observed by the Freja satellite: A statistical study*. Journal of Geophysical Research: Space Physics (1978–2012), 1998. **103**(A3): p. 4327-4341.
14. Haerendel, G., et al., *Inverted-V events simultaneously observed with the Freja satellite and from the ground*. Geophysical research letters, 1994. **21**(17): p. 1891-1894.
15. McFadden, J., et al., *Structure of plasmaspheric plumes and their participation in magnetopause reconnection: First results from THEMIS*. Geophysical Research Letters, 2008. **35**(17).
16. *THEMIS satellites*. 2014; Available from: http://www.nasa.gov/images/content/164101main_auroral_disruption_med.jpg.
17. Vallado, D.A. and W.D. McClain, *Fundamentals of Astrodynamics and Applications*. 2007: Springer.

18. Gurfil, P., *Modern Astrodynamics*. 2006: Elsevier Science.
19. Bate, R.R., D.D. Mueller, and J.E. White, *Fundamentals of Astrodynamics*. 1971: Dover Publications.
20. Dormand, J.R. and P.J. Prince, *A family of embedded Runge-Kutta formulae*. Journal of computational and applied mathematics, 1980. **6**(1): p. 19-26.
21. Kennedy, J. and R. Eberhart, *Particle swarm optimization: Proceedings of IEEE international conference on neural networks*. Perth, WA, 1995. **27**: p. 1.
22. Angeline, P.J. *Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences*. in *Evolutionary Programming VII*. 1998. Springer.
23. Birge, B. *PSOt-a particle swarm optimization toolbox for use with Matlab*. in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. 2003. IEEE.
24. Eberhart, R.C. and J. Kennedy. *A new optimizer using particle swarm theory*. in *Proceedings of the sixth international symposium on micro machine and human science*. 1995. New York, NY.
25. Pontani, M. and B.A. Conway, *Particle swarm optimization applied to space trajectories*. Journal of Guidance, Control, and Dynamics, 2010. **33**(5): p. 1429-1441.
26. Zhu, K.-J., J.-F. Li, and H.-X. Baoyin, *Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost*. Acta Astronautica, 2010. **66**(1): p. 220-229.
27. Lee, S., et al., *A statistical study of plasma sheet electrons carrying auroral upward field-aligned currents measured by Time History of Events and Macroscale Interactions during Substorms (THEMIS)*. Journal of Geophysical Research: Space Physics (1978–2012), 2011. **116**(A12).
28. Macmillan, S. and C. Finlay, *The International Geomagnetic Reference Field*, in *Geomagnetic Observations and Models*. 2011, Springer. p. 265-276.
29. Shi, Y. and R.C. Eberhart. *Parameter selection in particle swarm optimization*. in *Evolutionary Programming VII*. 1998. Springer.

Appendices

Appendix A. Tables of Optimal Solutions

Table A.1: Single burn optimization for satellite 1 with $\beta=0$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.3330	2.1313	0.0000
0.1	0.0000	0.3330	2.1313	0.0014
0.2	0.0000	0.3330	2.1313	0.0028
0.3	0.0000	0.3330	2.1313	0.0042
0.4	0.1046	0.8338	1.7248	0.0083
0.5	0.1046	0.8338	1.7248	0.0127
0.6	0.1046	0.8338	1.7248	0.0171
0.7	0.1046	0.8338	1.7248	0.0215
0.8	0.1046	0.8338	1.7248	0.0259
0.9	0.1046	0.8338	1.7248	0.0303
1	0.1048	0.8339	1.6928	0.0347

Table A.2: Single burn optimization for satellite 1 with $\beta=0.5$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0157	0.2436	1.0998	-0.0320
0.1	-0.0157	0.2436	1.0998	-0.0308
0.2	0.0307	0.3703	1.0838	-0.0292
0.3	0.0307	0.3703	1.0838	-0.0274
0.4	0.1053	0.8289	1.1817	-0.0246
0.5	0.1053	0.8289	1.1817	-0.0203
0.6	0.1053	0.8289	1.1817	-0.0159
0.7	0.1053	0.8289	1.1817	-0.0115
0.8	0.1053	0.8289	1.1817	-0.0071
0.9	0.1053	0.8289	1.1817	-0.0027
1	0.1053	0.8289	1.1817	0.0017

Table A.3: Single burn optimization for satellite 2 with $\beta=0$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.3330	2.1313	0.0000
0.1	0.0000	0.3330	2.1313	0.0014
0.2	0.0000	0.3330	2.1313	0.0028
0.3	-0.1108	0.9694	1.4106	0.0052
0.4	-0.1108	0.9694	1.4106	0.0102
0.5	-0.1108	0.9694	1.4106	0.0152
0.6	-0.1108	0.9694	1.4106	0.0203
0.7	-0.1108	0.9694	1.4106	0.0253
0.8	-0.1108	0.9694	1.4106	0.0303
0.9	-0.1108	0.9694	1.4106	0.0354
1	-0.1108	0.9694	1.4106	0.0404

Table A.4: Single burn optimization for satellite 2 with $\beta=0.5$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0319	0.3748	1.0705	-0.0326
0.1	-0.0319	0.3748	1.0705	-0.0307
0.2	-0.0319	0.3748	1.0705	-0.0289
0.3	-0.0319	0.3748	1.0705	-0.0270
0.4	-0.0319	0.3748	1.0705	-0.0252
0.5	-0.0323	0.3761	1.0708	-0.0234
0.6	-0.1260	0.9473	1.2399	-0.0153
0.7	-0.1260	0.9473	1.2399	-0.0102
0.8	-0.1260	0.9473	1.2399	-0.0051
0.9	-0.1260	0.9473	1.2399	0.0000
1	-0.1260	0.9473	1.2399	0.0050

Table A.5: Double burn optimization with $\beta=0$.

α	ΔV_1 , km/s	ΔV_2 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.0000	0.3330	2.1313	0.0000
0.1	0.0000	0.0000	0.3330	2.1313	0.0014
0.2	0.0000	0.0000	0.3330	2.1313	0.0028
0.3	0.0008	-0.1105	0.9932	1.9864	0.0055
0.4	0.0008	-0.1105	0.9933	1.9861	0.0106
0.5	0.0008	-0.1106	0.9933	1.9854	0.0157
0.6	0.0008	-0.1106	0.9933	1.9854	0.0209
0.7	-0.5587	-0.7009	3.5525	0.9932	0.0699
0.8	-0.5622	-0.7045	3.5563	0.9636	0.0959
0.9	-0.5622	-0.7045	3.5563	0.9636	0.1221
1	-0.5651	-0.7075	3.5575	0.9936	0.1482

Table A.6: Double burn optimization with $\beta=0.5$.

α	ΔV_1 , km/s	ΔV_2 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0157	0.0000	0.2436	1.0998	-0.0320
0.1	0.0000	-0.0319	0.3748	1.0705	-0.0307
0.2	0.0000	-0.0319	0.3748	1.0705	-0.0289
0.3	0.0084	-0.0236	0.3747	1.0653	-0.0269
0.4	-0.0090	-0.1193	0.9853	1.0457	-0.0195
0.5	-0.4758	-0.6161	2.9724	0.4518	0.0006
0.6	-0.4768	-0.6166	3.0420	0.4725	0.0239
0.7	-0.4768	-0.6166	3.0428	0.4733	0.0463
0.8	-0.4768	-0.6166	3.0431	0.4736	0.0688
0.9	-0.5618	-0.7025	3.4687	0.6943	0.0995
1	-0.5619	-0.7025	3.4551	0.6652	0.1255

Table A.7: Four burn optimization with $\beta=0$.

α	ΔV_1 , km/s	ΔV_2 , km/s	ΔV_3 , km/s	ΔV_4 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.0000	0.0000	0.0000	0.4961	1.9785	0.0000
0.1	0.0107	-0.0081	0.0000	0.0000	0.5677	2.1374	0.0022
0.2	0.3545	-0.9120	0.1146	-0.1884	2.3484	1.5990	0.0096
0.3	0.0199	-1.2709	0.2787	-0.0001	2.4754	1.5400	0.0222
0.4	0.0184	-1.2723	0.2787	0.0000	2.4756	1.5749	0.0338
0.5	-6.1726	-7.7502	0.1027	0.1028	6.9704	1.1551	0.0889
0.6	-6.1847	-7.7688	0.1263	0.1411	6.9753	1.1138	0.1290
0.7	-6.2981	-7.8885	0.0000	0.0212	7.1465	1.0005	0.1745
0.8	-6.2695	-7.8592	0.0000	0.0203	7.1331	0.9812	0.2152
0.9	-6.5495	-8.1448	0.5194	0.5753	6.8059	0.9881	0.2426
1	-5.9941	-7.5625	0.0002	0.0000	6.8514	1.2130	0.2855

Table A.8: Four burn optimization with $\beta=0.5$.

α	ΔV_1 , km/s	ΔV_2 , km/s	ΔV_3 , km/s	ΔV_4 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0172	0.0009	-0.0003	0.0007	0.4179	1.0520	-0.0309
0.1	-0.0165	0.0015	-0.0010	0.0001	0.4180	1.0521	-0.0290
0.2	0.0086	-0.0236	0.0009	0.0000	0.6345	1.0587	-0.0265
0.3	0.0058	-0.1106	0.0080	-0.0170	2.3743	1.1214	-0.0103
0.4	0.0244	-0.0891	0.0108	-0.0166	2.3420	1.1308	0.0001
0.5	-0.0257	-0.1420	0.0264	0.0028	2.4614	1.0285	0.0139
0.6	-0.0208	-0.1385	0.0061	-0.0177	2.4348	1.0611	0.0249
0.7	-0.5331	-0.6722	0.0909	0.0894	6.0711	0.8308	0.1169
0.8	-0.5059	-0.6439	0.0933	0.0902	5.7603	1.1511	0.1362
0.9	-0.5456	-0.6850	0.0339	0.0344	6.5705	0.6351	0.2172
1	-0.5977	-0.7408	0.1155	0.1254	6.1212	0.8109	0.2325

Table A.9: J₂ effect single burn optimization satellite 1 with $\beta=0$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.2727	1.4541	0.0000
0.1	-0.0003	0.2952	1.4541	0.0012
0.2	-0.0003	0.2952	1.4541	0.0024
0.3	-0.0003	0.2952	1.4541	0.0037
0.4	0.0945	0.7467	2.0795	0.0074
0.5	0.0945	0.7467	2.0795	0.0113
0.6	0.0945	0.7467	2.0795	0.0153
0.7	0.0945	0.7467	2.0795	0.0192
0.8	0.0945	0.7467	2.0795	0.0232
0.9	0.0945	0.7467	2.0795	0.0272
1	0.0945	0.7467	2.0795	0.0311

Table A.10: J₂ effect single burn optimization satellite 1 with $\beta=0.5$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0148	0.3235	1.2918	-0.0372
0.1	-0.0148	0.3235	1.2918	-0.0357
0.2	-0.0148	0.3235	1.2918	-0.0342
0.3	0.0468	0.4173	1.2582	-0.0327
0.4	0.0468	0.4173	1.2582	-0.0305
0.5	0.0978	0.7027	1.3576	-0.0274
0.6	0.0978	0.7027	1.3576	-0.0236
0.7	0.0978	0.7027	1.3576	-0.0198
0.8	0.0978	0.7027	1.3576	-0.0160
0.9	0.0978	0.7028	1.3576	-0.0122
1	0.0978	0.7028	1.3576	-0.0084

Table A.11: J₂ effect single burn optimization satellite 2 with $\beta=0$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.2727	1.4541	0.0000
0.1	0.0000	0.2727	1.4541	0.0011
0.2	0.0000	0.2727	1.4541	0.0023
0.3	-0.1005	0.8304	1.4686	0.0041
0.4	-0.1005	0.8304	1.4686	0.0085
0.5	-0.1005	0.8304	1.4686	0.0128
0.6	-0.1005	0.8304	1.4686	0.0172
0.7	-0.1005	0.8304	1.4686	0.0215
0.8	-0.1005	0.8304	1.4686	0.0259
0.9	-0.1005	0.8304	1.4686	0.0302
1	-0.1005	0.8304	1.4686	0.0346

Table A.12: J₂ effect single burn optimization satellite 2 with $\beta=0.5$.

α	ΔV , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0224	0.3173	1.2609	-0.0370
0.1	0.0224	0.3173	1.2609	-0.0355
0.2	0.0224	0.3173	1.2609	-0.0340
0.3	-0.0976	0.7918	1.2240	-0.0302
0.4	-0.0976	0.7919	1.2240	-0.0260
0.5	-0.0976	0.7920	1.2241	-0.0219
0.6	-0.0976	0.7921	1.2241	-0.0177
0.7	-0.0976	0.7921	1.2242	-0.0135
0.8	-0.0977	0.7922	1.2242	-0.0093
0.9	-0.0977	0.7922	1.2243	-0.0052
1	-0.0977	0.7923	1.2243	-0.0010

Table A.13: J₂ effect double burn optimization with $\beta=0$.

α	ΔV_1 , km/s	ΔV_2 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.0000	0.2727	1.4541	0.0000
0.1	-0.0003	0.0000	0.2952	1.4541	0.0012
0.2	-0.0003	0.0000	0.2952	1.4541	0.0024
0.3	0.0023	-0.0986	0.8646	1.4798	0.0045
0.4	0.0023	-0.0986	0.8646	1.4799	0.0090
0.5	0.0023	-0.0986	0.8646	1.4799	0.0135
0.6	0.0023	-0.0986	0.8646	1.4799	0.0180
0.7	-0.6664	-0.8000	3.7052	0.7878	0.0688
0.8	-0.6664	-0.8000	3.7052	0.7878	0.0973
0.9	-0.7110	-0.8465	3.9362	0.4971	0.1337
1	-0.6936	-0.8284	3.8563	0.5207	0.1607

Table A.14: J₂ effect double burn optimization with $\beta=0.5$.

α	ΔV_1 , km/s	ΔV_2 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0018	0.0143	0.3169	1.2505	-0.0362
0.1	-0.0018	0.0144	0.3169	1.2505	-0.0347
0.2	0.0130	-0.0114	0.3524	1.2695	-0.0341
0.3	0.0013	-0.0976	0.8273	1.1843	-0.0287
0.4	0.0104	-0.0895	0.8245	1.1974	-0.0249
0.5	-0.6981	-0.8330	3.8707	0.4953	-0.0015
0.6	-0.6901	-0.8248	3.8411	0.5394	0.0269
0.7	-0.6991	-0.8341	3.8728	0.4993	0.0580
0.8	-0.6660	-0.7997	3.7003	0.7435	0.0765
0.9	-0.6660	-0.7997	3.7003	0.7435	0.1050
1	-0.6938	-0.8285	3.8569	0.5193	0.1463

Table A.15: J₂ effect Four burn optimization with $\beta=0$.

α	ΔV_1 , km/s	ΔV_2 , km/s	ΔV_3 , km/s	ΔV_4 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	0.0000	0.0000	0.0000	0.0000	0.4625	1.3148	0.0000
0.1	-0.0003	0.0000	0.0000	0.0000	0.4842	1.5776	0.0020
0.2	0.0931	-0.0086	0.0036	-0.0308	2.1319	1.7525	0.0081
0.3	0.0780	-0.0231	0.0189	-0.0159	2.1628	1.4600	0.0185
0.4	0.0475	-0.0534	0.0362	-0.0001	2.1992	1.3526	0.0293
0.5	-0.0016	-0.1069	0.0303	-0.0004	2.3399	1.7331	0.0425
0.6	-0.6660	-0.8000	0.0679	0.0651	7.7231	0.5060	0.1360
0.7	-0.6365	-0.7686	0.0576	0.0543	7.0397	0.9399	0.1647
0.8	-0.6718	-0.8056	0.0357	0.0337	7.7068	0.5697	0.2293
0.9	-0.6544	-0.7877	0.0237	0.0206	7.6954	0.5998	0.2753
1	-0.6510	-0.7838	0.0607	0.0582	7.4095	0.9053	0.3087

Table A.16: J₂ effect Four burn optimization with $\beta=0.5$.

α	ΔV_1 , km/s	ΔV_2 , km/s	ΔV_3 , km/s	ΔV_4 , km/s	T_{event} , hr	Θ_{ave} , °	J
0	-0.0194	0.0085	-0.0009	0.0005	0.5436	1.1807	-0.0354
0.1	-0.0064	0.0000	0.0000	0.0041	0.5200	1.2341	-0.0330
0.2	-0.0064	0.0000	0.0000	0.0041	0.5200	1.2341	-0.0307
0.3	-0.0041	-0.1101	0.0303	0.0000	2.2937	1.1020	-0.0110
0.4	-0.0095	-0.1172	0.0334	0.0068	2.2716	1.1049	-0.0018
0.5	0.0304	-0.0744	-0.0038	-0.0356	2.1987	1.1119	0.0085
0.6	-0.6686	-0.8030	0.0482	0.0460	7.7581	0.4010	0.1269
0.7	-0.6371	-0.7698	0.0954	0.0951	6.8815	0.8293	0.1349
0.8	-0.6672	-0.8014	0.0733	0.0707	7.7737	0.4069	0.2190
0.9	-0.7130	-0.8482	0.0940	0.0949	7.0438	0.6003	0.2318
1	-0.6383	-0.7703	0.1271	0.1248	6.6480	0.8567	0.2532

Appendix B. PSO MATLAB Code

The following is the MATLAB code used to generate all the optimal solutions. See Reference [23] for more information. Code can also be found online at:

<http://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox>.

```
% pso_Trelea_vectorized.m
% a generic particle swarm optimizer
% to find the minimum or maximum of any
% MISO matlab function
%
% Implements Common, Trelea type 1 and 2, and Clerc's class 1". It will
% also automatically try to track to a changing environment (with varied
% success - BKB 3/18/05)
%
% This vectorized version removes the for loop associated with particle
% number. It also *requires* that the cost function have a single input
% that represents all dimensions of search (i.e., for a function that has 2
% inputs then make a wrapper that passes a matrix of ps x 2 as a single
% variable)
%
% Usage:
% [optOUT]=PSO(funcname,D)
% or:
% [optOUT,tr,te]=...
%     PSO(funcname,D,mv,VarRange,minmax,PSOparams,plotfcn,PSOseedValue)
%
% Inputs:
% funcname - string of matlab function to optimize
% D - # of inputs to the function (dimension of problem)
%
% Optional Inputs:
% mv - max particle velocity, either a scalar or a vector of length D
%      (this allows each component to have it's own max velocity),
%      default = 4, set if not input or input as NaN
%
% VarRange - matrix of ranges for each input variable,
% default -100 to 100, of form:
% [ min1 max1
%   min2 max2
```

```

%      ...
%      minD maxD ]
%
%      minmax = 0, funct minimized (default)
%      = 1, funct maximized
%      = 2, funct is targeted to P(12) (minimizes distance to errgoal)
%
%      PSOParams - PSO parameters
%      P(1) - Epochs between updating display, default = 100. if 0,
%             no display
%      P(2) - Maximum number of iterations (epochs) to train, default = 2000.
%      P(3) - population size, default = 24
%
%      P(4) - acceleration const 1 (local best influence), default = 2
%      P(5) - acceleration const 2 (global best influence), default = 2
%      P(6) - Initial inertia weight, default = 0.9
%      P(7) - Final inertia weight, default = 0.4
%      P(8) - Epoch when inertial weight at final value, default = 1500
%      P(9)- minimum global error gradient,
%             if abs(Gbest(i+1)-Gbest(i)) < gradient over
%             certain length of epochs, terminate run, default = 1e-25
%      P(10)- epochs before error gradient criterion terminates run,
%             default = 150, if the SSE does not change over 250 epochs
%             then exit
%      P(11)- error goal, if NaN then unconstrained min or max, default=NaN
%      P(12)- type flag (which kind of PSO to use)
%             0 = Common PSO w/inertia (default)
%             1,2 = Trelea types 1,2
%             3 = Clerc's Constricted PSO, Type 1"
%      P(13)- PSOseed, default=0
%             = 0 for initial positions all random
%             = 1 for initial particles as user input
%
%      plotfcn - optional name of plotting function, default 'goplotpso',
%             make your own and put here
%
%      PSOseedValue - initial particle position, depends on P(13), must be
%             set if P(13) is 1 or 2, not used for P(13)=0, needs to
%             be nXm where n<=ps, and m<=D
%             If n<ps and/or m<D then remaining values are set random
%             on Varrange
%      Outputs:
%      optOUT - optimal inputs and associated min/max output of function, of form:
%      [ bestin1
%        bestin2
%        ...

```



```

%      bestinD
%      bestOUT ]
%
% Optional Outputs:
%   tr  - Gbest at every iteration, traces flight of swarm
%   te  - epochs to train, returned as a vector 1:endepoch
%
% Example: out=pso_Trelea_vectorized('f6',2)

% Brian Birge
% Rev 3.3
% 2/18/06

function [OUT,varargout]=pso_Trelea_vectorized(funcname,D,varargin)

rand('state',sum(100*clock));
if nargin < 2
    error('Not enough arguments.');
```

end

```

% PSO PARAMETERS
if nargin == 2    % only specified funcname and D
    VRmin=ones(D,1)*-100;
    VRmax=ones(D,1)*100;
    VR=[VRmin,VRmax];
    minmax = 0;
    P = [];
    mv = 4;
    plotfcn='goplotpso';
elseif nargin == 3 % specified funcname, D, and mv
    VRmin=ones(D,1)*-100;
    VRmax=ones(D,1)*100;
    VR=[VRmin,VRmax];
    minmax = 0;
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    P = [];
    plotfcn='goplotpso';
elseif nargin == 4 % specified funcname, D, mv, Varrange
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    VR=varargin{2};
```

```

    minmax = 0;
    P = [];
    plotfcn='goplotpso';
elseif nargin == 5 % Functname, D, mv, Varrange, and minmax
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    VR=varargin{2};
    minmax=varargin{3};
    P = [];
    plotfcn='goplotpso';
elseif nargin == 6 % Functname, D, mv, Varrange, minmax, and psoparams
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    VR=varargin{2};
    minmax=varargin{3};
    P = varargin{4}; % psoparams
    plotfcn='goplotpso';
elseif nargin == 7 % Functname, D, mv, Varrange, minmax, and psoparams, plotfcn
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    VR=varargin{2};
    minmax=varargin{3};
    P = varargin{4}; % psoparams
    plotfcn = varargin{5};
elseif nargin == 8 % Functname, D, mv, Varrange, minmax, and psoparams, plotfcn,
PSOseedValue
    mv=varargin{1};
    if isnan(mv)
        mv=4;
    end
    VR=varargin{2};
    minmax=varargin{3};
    P = varargin{4}; % psoparams
    plotfcn = varargin{5};
    PSOseedValue = varargin{6};
else
    error('Wrong # of input arguments.');
```

end

% sets up default pso params

```

Pdef = [100 2000 24 2 2 0.9 0.4 1500 1e-25 250 NaN 0 0];
Plen = length(P);
P = [P,Pdef(Plen+1:end)];

df = P(1);
me = P(2);
ps = P(3);
ac1 = P(4);
ac2 = P(5);
iw1 = P(6);
iw2 = P(7);
iwe = P(8);
ergrd = P(9);
ergrdep = P(10);
errgoal = P(11);
trelea = P(12);
PSOseed = P(13);

% used with trainpso, for neural net training
if strcmp(funcname,'pso_neteval')
    net = evalin('caller','net');
    Pd = evalin('caller','Pd');
    TI = evalin('caller','TI');
    Ai = evalin('caller','Ai');
    Q = evalin('caller','Q');
    TS = evalin('caller','TS');
end

% error checking
if ((minmax==2) & isnan(errgoal))
    error('minmax= 2, errgoal= NaN: choose an error goal or set minmax to 0 or 1');
end

if ( ( PSOseed==1) & ~exist('PSOseedValue') )
    error('PSOseed flag set but no PSOseedValue was input');
end

if exist('PSOseedValue')
    tmpsz=size(PSOseedValue);
    if D < tmpsz(2)
        error('PSOseedValue column size must be D or less');
    end
    if ps < tmpsz(1)
        error('PSOseedValue row length must be # of particles or less');
    end
end

```

```

end

% set plotting flag
if (P(1))~=0
    plotflg=1;
else
    plotflg=0;
end

% preallocate variables for speed up
tr = ones(1,me)*NaN;

% take care of setting max velocity and position params here
if length(mv)==1
    velmaskmin = -mv*ones(ps,D); % min vel, psXD matrix
    velmaskmax = mv*ones(ps,D); % max vel
elseif length(mv)==D
    velmaskmin = repmat(forcerow(-mv),ps,1); % min vel
    velmaskmax = repmat(forcerow( mv),ps,1); % max vel
else
    error('Max vel must be either a scalar or same length as prob dimension D');
end
posmaskmin = repmat(VR(1:D,1)',ps,1); % min pos, psXD matrix
posmaskmax = repmat(VR(1:D,2)',ps,1); % max pos
posmaskmeth = 3; % 3=bounce method (see comments below inside epoch loop)

% PLOTTING
message = sprintf('PSO: %%g/%%g iterations, GBest = %%20.20g.\n',me);

% INITIALIZE INITIALIZE INITIALIZE INITIALIZE INITIALIZE INITIALIZE

% initialize population of particles and their velocities at time zero,
% format of pos= (particle#, dimension)
% construct random population positions bounded by VR
pos(1:ps,1:D) = normmat(rand([ps,D]),VR',1);

if PSOseed == 1 % initial positions user input, see comments above
    tmpsz = size(PSOseedValue);
    pos(1:tmpsz(1),1:tmpsz(2)) = PSOseedValue;
end

% construct initial random velocities between -mv,mv
vel(1:ps,1:D) = normmat(rand([ps,D]),...
    [forcecol(-mv),forcecol(mv)]',1);

% initial pbest positions vals

```

```

pbest = pos;

% VECTORIZE THIS, or at least vectorize cost funct call
out = feval(funcname,pos); % returns column of cost values (1 for each particle)
%-----

pbestval=out; % initially, pbest is same as pos

% assign initial gbest here also (gbest and gbestval)
if minmax==1
    % this picks gbestval when we want to maximize the function
    [gbestval,idx1] = max(pbestval);
elseif minmax==0
    % this works for straight minimization
    [gbestval,idx1] = min(pbestval);
elseif minmax==2
    % this works when you know target but not direction you need to go
    % good for a cost function that returns distance to target that can be either
    % negative or positive (direction info)
    [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal).^2);
    gbestval = pbestval(idx1);
end

% preallocate a variable to keep track of gbest for all iters
bestpos = zeros(me,D+1)*NaN;
gbest = pbest(idx1,:); % this is gbest position
% used with trainpso, for neural net training
% assign gbest to net at each iteration, these interim assignments
% are for plotting mostly
if strcmp(funcname,'pso_neteval')
    net=setx(net,gbest);
end
%tr(1) = gbestval; % save for output
bestpos(1,1:D) = gbest;

% this part used for implementing Carlisle and Dozier's APSO idea
% slightly modified, this tracks the global best as the sentry whereas
% their's chooses a different point to act as sentry
% see "Tracking Changing Extrema with Adaptive Particle Swarm Optimizer",
% part of the WAC 2002 Proceedings, June 9-13, http://wacong.com
sentryval = gbestval;
sentry = gbest;

if (trelea == 3)
% calculate Clerc's constriction coefficient chi to use in his form
kappa = 1; % standard val = 1, change for more or less constriction

```

```

if ( (ac1+ac2) <=4 )
    chi = kappa;
else
    psi    = ac1 + ac2;
    chi_den = abs(2-psi-sqrt(psi^2 - 4*psi));
    chi_num = 2*kappa;
    chi    = chi_num/chi_den;
end
end

% INITIALIZE END INITIALIZE END INITIALIZE END INITIALIZE END
rstflg = 0; % for dynamic environment checking
% start PSO iterative procedures
cnt    = 0; % counter used for updating display according to df in the options
cnt2   = 0; % counter used for the stopping subroutine based on error convergence
iwt(1) = iwl;
for i=1:me % start epoch loop (iterations)

    out    = feval(funcname,[pos;gbest]);
    outbestval = out(end,:);
    out    = out(1:end-1,:);

    tr(i+1)    = gbestval; % keep track of global best val
    te        = i; % returns epoch number to calling program when done
    bestpos(i,1:D+1) = [gbest,gbestval];

    %assignin('base','bestpos',bestpos(i,1:D+1));
%-----
% this section does the plots during iterations
if plotflg==1
    if (rem(i,df) == 0 ) | (i==me) | (i==1)
        fprintf(message,i,gbestval);
        cnt = cnt+1; % count how many times we display (useful for movies)

        eval(plotfcn); % defined at top of script

    end % end update display every df if statement
end % end plotflg if statement

% check for an error space that changes wrt time/iter
% threshold value that determines dynamic environment
% sees if the value of gbest changes more than some threshold value
% for the same location
chkdyn = 1;
rstflg = 0; % for dynamic environment checking

```

```

if chkdyn==1
    threshld = 0.05; % percent current best is allowed to change, .05 = 5% etc
    letiter = 5; % # of iterations before checking environment, leave at least 3 so PSO has time to
converge
    outorng = abs( 1- (outbestval/gbestval) ) >= threshld;
    samepos = (max( sentry == gbest ));

    if (outorng & samepos) & rem(i,letiter)==0
        rstflg=1;
        % disp('New Environment: reset pbest, gbest, and vel');
        %% reset pbest and pbestval if warranted
    %     outpbestval = feval( functname,[pbest] );
    %     Poutorng = abs( 1-(outpbestval./pbestval) ) > threshld;
    %     pbestval = pbestval.*~Poutorng + outpbestval.*Poutorng;
    %     pbest = pbest.*repmat(~Poutorng,1,D) + pos.*repmat(Poutorng,1,D);

    pbest = pos; % reset personal bests to current positions
    pbestval = out;
    vel = vel*10; % agitate particles a little (or a lot)

    % recalculate best vals
    if minmax == 1
        [gbestval,idx1] = max(pbestval);
    elseif minmax==0
        [gbestval,idx1] = min(pbestval);
    elseif minmax==2 % this section needs work
        [temp,idx1] = min((pbestval-ones(size(pbestval))*errgoal).^2);
        gbestval = pbestval(idx1);
    end

    gbest = pbest(idx1,:);

    % used with trainpso, for neural net training
    % assign gbest to net at each iteration, these interim assignments
    % are for plotting mostly
    if strcmp(functname,'pso_neteval')
        net=setx(net,gbest);
    end
end % end if outorng

sentryval = gbestval;
sentry = gbest;

end % end if chkdyn

% find particles where we have new pbest, depending on minmax choice

```

```

% then find gbest and gbestval
%[size(out),size(pbestval)]
if rstflg == 0
if minmax == 0
    [tempi]      = find(pbestval>=out); % new min pbestvals
    pbestval(tempi,1) = out(tempi); % update pbestvals
    pbest(tempi,:)  = pos(tempi,:); % update pbest positions

    [iterbestval,idx1] = min(pbestval);

    if gbestval >= iterbestval
        gbestval = iterbestval;
        gbest    = pbest(idx1,:);
        % used with trainpso, for neural net training
        % assign gbest to net at each iteration, these interim assignments
        % are for plotting mostly
        if strcmp(funcname,'pso_neteval')
            net=setx(net,gbest);
        end
    end
elseif minmax == 1
    [tempi,dum]      = find(pbestval<=out); % new max pbestvals
    pbestval(tempi,1) = out(tempi,1); % update pbestvals
    pbest(tempi,:)  = pos(tempi,:); % update pbest positions

    [iterbestval,idx1] = max(pbestval);
    if gbestval <= iterbestval
        gbestval = iterbestval;
        gbest    = pbest(idx1,:);
        % used with trainpso, for neural net training
        % assign gbest to net at each iteration, these interim assignments
        % are for plotting mostly
        if strcmp(funcname,'pso_neteval')
            net=setx(net,gbest);
        end
    end
elseif minmax == 2 % this won't work as it is, fix it later
    egones      = errgoal*ones(ps,1); % vector of errgoals
    sqrrerr2     = ((pbestval-egones).^2);
    sqrrerr1     = ((out-egones).^2);
    [tempi,dum]  = find(sqrrerr1 <= sqrrerr2); % find particles closest to targ
    pbestval(tempi,1) = out(tempi,1); % update pbestvals
    pbest(tempi,:)  = pos(tempi,:); % update pbest positions

    sqrrerr      = ((pbestval-egones).^2); % need to do this to reflect new pbests
    [temp,idx1]   = min(sqrrerr);

```



```

        +1.494.*rannum2.*(repmat(gbest,ps,1)-pos); % social
elseif trelea == 1
% from Trelea's paper, parameter set 1
vel = 0.600.*vel... % prev vel
    +1.700.*rannum1.*(pbest-pos)... % independent
    +1.700.*rannum2.*(repmat(gbest,ps,1)-pos); % social
elseif trelea == 3
% Clerc's Type 1" PSO
vel = chi*(vel... % prev vel
    +ac1.*rannum1.*(pbest-pos)... % independent
    +ac2.*rannum2.*(repmat(gbest,ps,1)-pos)) ; % social
else
% common PSO algo with inertia wt
% get inertia weight, just a linear funct w.r.t. epoch parameter iwe
if i<=iwe
    iwt(i) = ((iw2-iw1)/(iwe-1))*(i-1)+iw1;
else
    iwt(i) = iw2;
end
% random number including acceleration constants
ac11 = rannum1.*ac1; % for common PSO w/inertia
ac22 = rannum2.*ac2;

%    veltemp=vel;
    vel = iwt(i).*vel... % prev vel
        +ac11.*(pbest-pos)... % independent
        +ac22.*(repmat(gbest,ps,1)-pos); % social
end

% limit velocities here using masking
vel = ( (vel <= velmaskmin).*velmaskmin ) + ( (vel > velmaskmin).*vel );
vel = ( (vel >= velmaskmax).*velmaskmax ) + ( (vel < velmaskmax).*vel );

% update new position (PSO algo)
%    postemp=pos;
    pos = pos + vel;

%
% position masking, limits positions to desired search space
% method: 0) no position limiting, 1) saturation at limit,
%    2) wraparound at limit , 3) bounce off limit
minposmask_throwaway = pos <= posmaskmin; % these are psXD matrices
minposmask_keep = pos > posmaskmin;
maxposmask_throwaway = pos >= posmaskmax;
maxposmask_keep = pos < posmaskmax;

```



```

    if plotflg == 1
        fprintf(message,i,gbestval);
        disp(' ');
        disp(['--> Error Goal reached, successful termination!']);

        eval(plotfcn);
    end
    break
end

% this is stopping criterion for constrained from both sides
if minmax == 2
    if ((tr(i)<errgoal) & (gbestval>=errgoal)) | ((tr(i)>errgoal) ...
        & (gbestval <= errgoal))
        if plotflg == 1
            fprintf(message,i,gbestval);
            disp(' ');
            disp(['--> Error Goal reached, successful termination!']);

            eval(plotfcn);
        end
        break
    end
end % end if minmax==2
end % end ~isnan if

% % convert back to inertial frame
% pos = pos - repmat(gbestoffset,ps,1);
% pbest = pbest - repmat(gbestoffset,ps,1);
% gbest = gbest + gbestoffset;

end % end epoch loop

%% clear temp outputs
% evalin('base','clear temp_pso_out temp_te temp_tr;');

% output & return
OUT=[gbest';gbestval];
varargout{1}=[1:te];
varargout{2}=[tr(find(~isnan(tr)))];

return

```